JUNOS™ Internet Software
for J-series™, M-series™, and T-series™
Routing Platforms

Network Management Configuration Guide

Release 7.4

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

Part Number: 530-014069-01, Revision 1

JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms Network Management Configuration Guide, Release 7.4
Writing: Lisa Kelly
Editing: Sonia Saruba
Illustration: Nathaniel Woodward
Cover design: Edmonds Design

Revision History
14 September 2005—Revision 1

Year 2000 Notice

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

Software License

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use the Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller, unless the applicable Juniper documentation expressly permits installation on non-Juniper equipment.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Software on non-Juniper equipment where the Juniper documentation does not expressly permit installation on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; or (k) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Interface Information. To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. Third Party Software. Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

Part 6 Juniper Networks Enterprise-Specific MIBs

About This Guide

This preface provides the following guidelines for using the *JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms Network Management Configuration Guide* and related Juniper Networks, Inc., technical documents:

Objectives

This guide provides an overview of the network management features of the JUNOS Internet software and describes how to manage your network with the JUNOS software.

NOTE: This guide documents Release 7.4 of the JUNOS Internet software. For additional information about the JUNOS software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks J-series, M-series, or T-series routing platform.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Using the Indexes

This guide contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines*, refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

Documentation Conventions

Table 1 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.

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

Table 2: Text and Syntax Conventions

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

Convention	Element	Example

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold typeface	Represents J-Web graphical user interface (GUI) items you click or select.	■ In the Logical Interfaces box, select **All Interfaces**. ■ To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols > Ospf**.

Related Juniper Networks Documentation

Table 3 lists the software and hardware guides and release notes for Juniper Networks J-series, M-series, and T-series routing platforms and describes the contents of each document. Table 4 lists the books included in the *Network Operations Guide* series.

Table 3: Technical Documentation for J-series, M-series, and T-series Routing Platforms

Document	Description
JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms Configuration Guides	
Class of Service	Provides an overview of the class-of-service (CoS) functions of the JUNOS software and describes how to configure CoS features, including configuring multiple forwarding classes for transmitting packets, defining which packets are placed into each output queue, scheduling the transmission service level for each queue, and managing congestion through the random early detection (RED) algorithm.
Feature Guide	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
JUNOS-FIPS	(M-series and T-series routing platforms only) Provides an overview of JUNOS-FIPS 140-2 concepts and describes how to install and configure the JUNOS-FIPS software. Describes FIPS-related commands and how to configure, authorize, and zeroize the Adaptive Services (AS) II FIPS Physical Interface Card (PIC).
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast Protocols	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
Network Interfaces	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, forwarding options, and cflowd.

Document	Description
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
Services Interfaces	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the routing platform.
System Basics	Provides an overview of the JUNOS software and describes how to install and upgrade the software. This guide also describes how to configure system management functions and how to configure the chassis, including user accounts, passwords, and redundancy.
VPNs	Provides an overview and describes how to configure Layer 2 and Layer 3 virtual private networks (VPNs), virtual private LAN service (VPLS), and Layer 2 circuits. Provides configuration examples.
JUNOS References	
Interfaces Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
Routing Protocols and Policies Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing protocols and policies, including firewall filters.
System Basics and Services Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot system basics, including commands for real-time monitoring and route (or path) tracing, system software management, and chassis management. Also describes commands for monitoring and troubleshooting services such as CoS, IP Security (IPSec), stateful firewalls, flow collection, and flow monitoring.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
J-Web User Guide	
J-Web Interface User Guide	Describes how to use the J-Web GUI to configure, monitor, and manage Juniper Networks routing platforms.
JUNOS API and Scripting Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
JUNOScript API Configuration Reference	Provides a reference page for the configuration tags in the JUNOScript API.
JUNOScript API Operational Reference	Provides a reference page for the operational tags in the JUNOScript API.
JUNOS Configuration Scripting Guide	Provides an overview, instructions for using, and examples of the commit script feature of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts that run at commit time. This guide also explains how to use commit script macros to provide simplified aliases for frequently used configuration statements.
JUNOS Comprehensive Index and Glossary	
Comprehensive Index and Glossary	Provides a complete index of all JUNOS software books and the *JUNOScript API Guide*. Also provides a comprehensive glossary.
JUNOScope Documentation	
JUNOScope Software User Guide	Describes the JUNOScope software GUI, how to install and administer the software, and how to use the software to manage routing platform configuration files and monitor routing platform operations.

Document	Description
J-series Services Router Documentation	
J-series Services Router Getting Started Guide	Provides an overview, basic instructions, and specifications for J-series Services Routers. The guide explains how to prepare your site for installation, unpack and install the router and its components, install licenses, and establish basic connectivity.
J-series Services Router Configuration Guide	Explains how to configure the interfaces on J-series Services Routers for basic IP routing with standard routing protocols. The guide also shows how to configure VPNs, configure and manage multicast networks, and apply routing techniques such as policies, firewall filters, IPSec tunnels, and service classification for safer, more efficient routing.
J-series Services Router Administration Guide	Shows how to manage users and operations, monitor network performance, upgrade software, and diagnose common problems on J-series Services Routers.
M-series and T-series Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
PIC Guide	Describes the routing platform PICs. Each platform has its own PIC guide.
Release Notes	
JUNOS Release Notes	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS and JUNOScript manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
Hardware Release Notes	Describe the available documentation for the routing platform and the supported PICs, and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
JUNOScope Software Release Notes	Contain corrections and updates to the published JUNOScope manual, provide information that might have been omitted from the manual, and describe upgrade and downgrade procedures.
J-series Services Router Release Notes	Briefly describe the J-series Services Router features, identify known hardware problems, and provide upgrade and downgrade instructions.

Table 4: JUNOS Internet Software Network Operations Guides

Book	Description
JUNOS Internet Software for M-series and T-series Routing Platforms Network Operations Guides	
Baseline	Describes the most basic tasks for running a network using Juniper Networks products. Tasks include upgrading and reinstalling JUNOS software, gathering basic system management information, verifying your network topology, and searching log messages.
Interfaces	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
MPLS	Describes tasks for configuring, monitoring, and troubleshooting an example MPLS network. Tasks include verifying the correct configuration of the MPLS and RSVP protocols, displaying the status and statistics of MPLS running on all routers in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.

Book	Description
MPLS Log Reference	Describes MPLS status and error messages that appear in the output of the **show mpls lsp extensive** command. The guide also describes how and when to configure Constrained Shortest Path First (CSPF) and RSVP trace options, and how to examine a CSPF or RSVP failure in a sample network.
Hardware	Describes tasks for monitoring M-series and T-series routing platforms.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at http://www.juniper.net/techpubs/docbug/docbugreport.html. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version

Requesting Support

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1
Network Management Introduction

Chapter 1

Network Management Overview

Once you have installed the router into your network, you need to manage the router within your network. Router management can be divided into five tasks:

- Fault management—Monitor the router; detect and fix faults.
- Configuration management—Configure router attributes.
- Accounting management—Collect statistics for accounting purposes.
- Performance management—Monitor and adjust router performance.
- Security management—Control router access and authenticate users.

The JUNOS software network management features work in conjunction with an operations support system (OSS) to manage the router within the network. The JUNOS software can assist you in performing these management tasks, as described in Table 5.

Table 5: JUNOS Router Management Features

Task	JUNOS Software Feature
Fault management	Monitor and see faults using: ■ Operational mode commands—For more information on operational mode commands, see the *JUNOS System Basics and Services Command Reference, JUNOS Interfaces Command Reference,* and *JUNOS Routing Protocols and Policies Command Reference.* ■ SNMP MIBs—For more information about SNMP MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 105. ■ Standard SNMP traps—For more information about standard SNMP traps, see "Standard SNMP Traps" on page 119. ■ Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 111. ■ System log messages—For more information about how to configure system log messages, see the *JUNOS System Basics Configuration Guide.* For more information about how to view system log messages, see the *JUNOS System Log Messages Reference.*
Configuration management	■ Configure router attributes using the command-line interface (CLI) and the JUNOScript API. For more information on configuring the router using the CLI, see the *JUNOS System Basics Configuration Guide.* For more information on configuring the router using the JUNOScript API, see the *JUNOScript API References* and the *JUNOScript API Guide.* ■ Configuration Management MIB—For more information about the Configuration Management MIB, see "Juniper Networks Enterprise-Specific MIBs" on page 105.

Task	JUNOS Software Feature
Accounting management	Perform the following accounting-related tasks: ■ Collect statistics for interfaces, firewall filters, destination classes, source classes, and the Routing Engine. For more information on collecting statistics, see "Configuring Accounting Options" on page 399. ■ Use interface-specific traffic statistics and other counters, available in the standard interfaces MIB, Juniper Networks enterprise-specific extensions to the interfaces MIB, and media-specific MIBs, such as the enterprise-specific ATM MIB. ■ Use per-ATM virtual circuit (VC) counters, available in the enterprise-specific ATM MIB. ■ Group source and destination prefixes into source classes and destination classes and count packets for those classes. Collect destination class and source class usage statistics. For more information on classes, see "Juniper Networks Enterprise-Specific MIBs" on page 105,"Configuring Class Usage Profiles" on page 415, the *JUNOS Network Interfaces Configuration Guide*, and the *JUNOS Policy Framework Configuration Guide*. ■ Count packets as part of a firewall filter. For more information on firewall filter policies, see "Juniper Networks Enterprise-Specific MIBs" on page 105 and the *JUNOS Policy Framework Configuration Guide*. ■ Sample traffic, collect the samples, and send the collection to a host running the CAIDA cflowd utility. For more information on CAIDA and cflowd, see the *JUNOS Policy Framework Configuration Guide*.
Performance management	Monitor performance in the following ways: Use operational mode commands. For more information on monitoring performance using operational mode commands, see the *JUNOS System Basics and Services Command Reference*. ■ As part of a firewall filter. For more information on performance monitoring using firewall filters, see the *JUNOS Configuration Policy Framework Guide*. ■ Sample traffic, collect the samples, and send the samples to a host running the CAIDA cflowd utility. For more information on CAIDA and cflowd, see the *JUNOS Policy Framework Configuration Guide*. ■ Use the enterprise-specific class-of-service MIB. For more information on this MIB, see "Juniper Networks Enterprise-Specific MIBs" on page 105.
Security management	Assure security in your network in the following ways: ■ Control access to the router and authenticate users. For more information on access control and user authentication, see the *JUNOS System Basics Configuration Guide*. ■ Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see "Configuring the Local Engine ID" on page 41 and "Tracing SNMP Activity" on page 40.

Chapter 2

Complete Network Management Configuration Statements

This chapter shows the complete configuration statement hierarchy for the portions of the configuration discussed in this manual, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the JUNOS software, your current hierarchy level is shown in the banner on the line preceding the user@host# prompt.

For a list of the complete configuration statement hierarchy, see the *JUNOS System Basics Configuration Guide*.

This chapter is organized as follows:

[edit accounting-options] Hierarchy Level

```
[edit]
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        destination-classes {
            destination-class-name;
        }
        source-classes {
            source-class-name;
        }
    }
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
    }
```

```
    filter-profile profile-name {
      counters {
        counter-name;
      }
      file filename;
      interval minutes;
      }
    }
    interface-profile profile-name {
      fields {
        field-name;
      }
      file filename;
      interval minutes;
    }
    routing-engine-profile profile-name {
      fields {
        field-name;
      }
      file filename;
      interval minutes;
    }
}
```

[edit snmp] Hierarchy Level

```
[edit]
snmp {
    community community-name {
      authorization authorization;
      clients {
        address restrict;
      }
      view view-name;
    }
    contact contact;
    description description;
    engine-id {
      (local engine-id | use-default-ip-address | use-mac-address);
    }
    filter-duplicates;
    interface [ interface-names ];
    location location;
    name name;
    nonvolatile {
      commit-delay seconds;
    }
    rmon {
      alarm index {
        description description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        rising-threshold integer;
        sample-type type;
```

```
        startup-alarm alarm;
        variable oid-variable;
    }
    event index {
        community community-name;
        description description;
        type type;
    }
}
traceoptions {
    file size size files number;
    flag flag;
}
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
        address;
    }
    version (all | v1 | v2);
}
trap-options {
        agent-address outgoing-interface;
        source-address address;
}
v3 {
    notify name {
        tag tag-name;
        type (trap | inform);
    }
    notify-filter profile-name {
        oid oid (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;
        inform-timeout number;
        inform-retry-count seconds;
        port <port-number>;
        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-model (usm | v1 | v2c);
                security-level (authentication | none | privacy);
                security-name security-name;
        }
    }
```

```
        usm {
            local-engine {
                user username {
                    authentication-md5 {
                        authentication-password authentication-password;
                    }
                    authentication-none;
                    authentication-sha {
                        authentication-password authentication-password;
                    }
                    privacy-3des {
                        privacy-password privacy-password;
                    }
                    privacy-aes128 {
                        privacy-password privacy-password;
                    }
                    privacy-des {
                        privacy-password privacy-password;
                    }
                    privacy-none;
                }
            }
        }
        vacm {
            access {
                group group-name {
                    default-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);
    }
}
```

Part 2
Interim Local Management Interface

Chapter 3

Interim Local Management Interface Overview

The Integrated Local Management Interface (ILMI), provides a mechanism for Asynchronous Transfer Mode (ATM)-attached devices, such as hosts, routers, and ATM switches, to transfer management information. The ILMI provides bidirectional exchange of management information between two ATM interfaces across a physical connection. ILMI information is exchanged over a direct encapsulation of Simple Network Management Protocol (SNMP) version 1 (RFC 1157, *A Simple Network Management Protocol* [SNMP]) over ATM Adaptation Layer 5 (AAL5) using a virtual path identifier/virtual channel identifier (VPI/VCI) value (VPI = 0, VCI = 16).

The JUNOS software supports only two ILMI Management Information Base (MIB) variables: atmfMYIPNmAddress and atmfPortMyIfname. For ATM1 and ATM2 intelligent queuing (IQ) interfaces, you can configure ILMI to communicate directly to an attached ATM switch to enable querying of the switch's IP address and port number.

For more information about configuring ILMI, see the *JUNOS Network Interfaces Configuration Guide*. For information about displaying ILMI statistics, see the *JUNOS Interfaces Command Reference*. For more information about the ILMI MIB, see the ATM Forum at http://www.atmforum.com/.

Part 3
SNMP

Chapter 4

SNMP Overview

The Simple Network Management Protocol (SNMP) enables the monitoring of network devices from a central location. This chapter provides an overview of SNMP and describes how SNMP is implemented in the JUNOS software.

This chapter covers the following topics:

SNMP Architecture

The SNMP agent exchanges network management information with SNMP manager software running on a network management system (NMS), or host. The agent responds to requests for information and actions from the manager. The agent also controls access to the agent's Management Information Base (MIB), the collection of objects that can be viewed or changed by the SNMP manager.

The SNMP manager collects information on network connectivity, activity, and events by polling managed devices.

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.

Management Information Base

A MIB, or Management Information Base, is a hierarchy of information used to define managed objects in a network device. 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.

MIBs are either standard or enterprise-specific. Standard MIBs are created by the Internet Engineering Task Force (IETF) and documented in various RFCs. Depending on the vendor, many standard MIBs are delivered with the NMS software. You can also download the standard MIBs from the IETF Web site, http://www.ietf.org, and compile them into your NMS, if necessary.

For a list of standard supported MIBS, see "SNMP Standards" on page 19.

Enterprise-specific MIBs are developed and supported by a specific equipment manufacturer. If your network contains devices that have enterprise-specific MIBs, you must obtain them from the manufacturer and compile them into your network management software.

For a list of Juniper Networks enterprise-specific supported MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 105.

SNMP Traps and Informs

Routers can send notifications to SNMP managers when significant events occur on a network device, most often errors or failures. SNMP notifications can be sent as traps or inform requests. SNMP traps are unconfirmed notifications. SNMP informs are confirmed notifications.

SNMP traps are defined in either standard or enterprise-specific MIBs. Standard traps are created by the IETF and documented in various RFCs. The standard traps are compiled into the network management software. You can also download the standard traps from the IETF Web site, http://www.ietf.org.

For more information on standard traps supported by the JUNOS software, see "Standard SNMP Traps" on page 119.

Enterprise-specific traps are developed and supported by a specific equipment manufacturer. If your network contains devices that have enterprise-specific traps, you must obtain them from the manufacturer and compile them into your network management software.

For more information on enterprise-specific traps supported by the JUNOS software, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 111. For information on system logging severity levels for SNMP traps, see "System Logging Severity Levels for SNMP Traps" on page 24.

With traps, the receiver does not send any acknowledgment when it receives a trap and the sender cannot determine if the trap was received. To increase reliability, SNMP informs are supported in SNMPv3. An SNMP manager that receives an inform acknowledges the message with a response. For information on SNMP informs, see "Configuring SNMP Informs" on page 72.

SNMP Standards

The following standards documents define SNMP and the standard MIBs supported by the JUNOS software. RFCs can be found at http://www.ietf.org.

- IEEE, 802.3ad, *Aggregation of Multiple Link Segments*

  Only the following are supported:

  - dot3adAggPortTable, dot3adAggPortListTable, dot3adAggTable, and dot3adAggPortStatsTable
  - dot3adAggPortDebugTable (only dot3adAggPortDebugRxState, dot3adAggPortDebugMuxState, dot3adAggPortDebugActorSyncTransitionCount, dot3adAggPortDebugPartnerSyncTransitionCount, dot3adAggPortDebugActorChangeCount, and dot3adAggPortDebugPartnerChangeCount)
  - dot3adTablesLastChanged

- RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*
- RFC 1157, *A Simple Network Management Protocol (SNMP)*
- RFC 1213, *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II*. The JUNOS software supports the following areas:
  - MIB II and its SNMPv2 derivatives, including:
    - Statistics counters
    - IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, *IP Forwarding Table MIB*)
    - SNMP management
    - Interface management
  - SNMPv1 Get, GetNext requests, and version 2 GetBulk request
  - JUNOS-specific secured access list
  - Master configuration keywords
  - Reconfigurations upon SIGHUP
- RFC 1215, *A Convention for Defining Traps for use with the SNMP* (only MIB II SNMP version 1 traps and version 2 notifications)
- RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*
- RFC 1850, *OSPF Version 2 Management Information Base* (except for the ospfOriginateNewLsas and ospfRxNewLsas objects, the Host Table, and the traps ospfOriginateLSA, ospfLsdbOverflow, and ospfLsdbApproachingOverflow)
- RFC 1901, *Introduction to Community-based SNMPv2*
- RFC 1905, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*
- RFC 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*
- RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2*
- RFC 2012, *SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2*
- RFC 2013, *SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*

- RFC 2096, *IP Forwarding Table MIB*
- RFC 2115, *Management Information Base for Frame Relay DTEs Using SMIv2*
- RFC 2662. *Definitions of Managed Objects for ADSL Lines* (J-Series Services Routers. All MIB tables, objects, and traps applicable for the ADSL ATU-R agent.)
- RFC 2287, *Definitions of System-Level Managed Objects for Applications* (only sysApplInstallPkgTable, sysApplInstallElmtTable, sysApplElmtRunTable, and sysApplMapTable)
- RFC 2465, *Management Information Base for IP Version 6: Textual Conventions and General Group* (except for IPv6 interface statistics)
- RFC 2495, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types* (except for dsx1FarEndConfigTable, dsx1FarEndCurrentTable, dsx1FarEndIntervalTable, dsx1FarEndTotalTable, and dsx1FracTable)
- RFC 2496, *Definitions of Managed Objects for the DS3/E3 Interface Type* (except dsx3FarEndConfigTable, dsx3FarEndCurrentTable, dsx3FarEndIntervalTable, dsx3FarEndTotalTable, and dsx3FracTable)
- RFC 2515, *Definitions of Managed Objects for ATM Management* (except atmVpCrossConnectTable, atmVcCrossConnectTable, and aal5VccTable)
- RFC 2558, *Definitions of Managed Objects for the SONET/SDH Interface Type*
- RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework*
- RFC 2571, *An Architecture for Describing SNMP Management Frameworks* (read-only access)
- RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* (read-only access)
- RFC 2576, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
- RFC 2578, *Structure of Management Information Version 2 (SMIv2)*
- RFC 2579, *Textual Conventions for SMIv2*
- RFC 2665, *Definitions of Managed Objects for the Ethernet-like Interface Types*
- RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

- RFC 2790, *Host Resources MIB*
  - Only the hrStorageTable. The file systems /, /config, /var, and /tmp will 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, *Remote Network Monitoring Management Information Base* (the etherStatsTable for Ethernet interfaces only and the objects alarmTable, eventTable, and logTable)
- RFC 2863, *The Interfaces Group MIB*
- RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations* (only the objects pingCtlTable, pingResultsTable, pingProbeHistoryTable, pingMaxConcurrentRequests, traceRouteCtlTable, traceRouteResultsTable, traceRouteProbeHistoryTable, and traceRouteHopsTable)
- RFC 2932, *IPv4 Multicast Routing MIB*
- RFC 3413, *Simple Network Management Protocol (SNMP) Applications* (except for the proxy MIB*)*
- RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
- RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
- RFC 3811, *Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management*
- RFC 3813. *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)* (read only access. mplsInterfacePerfTable, mplsInSegmentPerfTable, mplsOutSegmentPerfTable, mplsInSegmentMapTable, mplsXCUp, and mplsXCDown are not supported)
- Internet draft draft-blumenthal-aes-usm-08.txt, *The AES Cipher Algorithm in the SNMP User-based Security Model*
- Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version* (only jnxBgpM2PrefixInPrefixes, jnxBgpM2PrefixInPrefixesAccepted, and jnxBgpM2PrefixInPrefixesRejected objects)
- Internet draft draft-reeder-snmpv3-usm-3desede-00.txt, *Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode*
- Internet draft draft-ietf-atommib-sonetaps-mib-10.txt, *Definitions of Managed Objects for SONET Linear APS architectures* (as defined under the Juniper Networks enterprise branch only)

- Internet draft draft-ietf-idmr-igmp-mib-13.txt, *Internet Group Management Protocol (IGMP) MIB*
- Internet Assigned Numbers Authority, *IANAiftype Textual Convention MIB* (referenced by RFC 2233, available at ftp://ftp.isi.edu/mib/ianaiftype.mib)
- Internet draft draft-ietf-isis-wg-mib-07.txt, *Management Information Base for IS-IS*, (only isisISAdjTable, isisISAdjAreaAddrTable, isisISAdjIPAddrTable, and isisISAdjProtSuppTable)
- Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt, *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (only mplsVpnScalars, mplsVpnVrfTable, mplsVpnPerTable, and mplsVpnVrfRouteTargetTable)
- Internet draft draft-ietf-msdp-mib-07.txt, *Multicast Source Discovery protocol MIB* (except msdpEstablished, msdpBackwardTransition, and msdpRequestsTable)
- Internet draft draft-ietf-idmr-pim-mib-09.txt, *Protocol Independent Multicast (PIM) MIB*
- ESO Consortium MIB, which can be found at http://www.snmp.com/eso/

JUNOS SNMP Agent Features

The JUNOS SNMP agent software consists of an SNMP master agent that delegates all SNMP requests to subagents. Each subagent is responsible for the support of a specific set of MIBs.

The JUNOS software supports the following versions of SNMP:

- SNMPv1—The initial implementation of SNMP that defines the architecture and framework for SNMP.
- SNMPv2c—The revised protocol, with improvements to performance and manager-to-manager communications. Specifically, SNMPv2c implements community strings, which act as passwords when determining who, what, and how the SNMP clients can access the data in the SNMP agent. The community string is contained in SNMP Get, GetBulk, GetNext, and Set requests. The agent may require a different community string for Get, GetBulk, and GetNext requests (read-only access) than it does for Set requests (read-write access).
- SNMPv3—The most up-to-date protocol focuses on security. SNMPv3 defines a security model, user-based security model (USM), and a view-based access control model (VACM). SNMPv3 USM provides data integrity, data origin authentication, message replay protection, and protection against disclosure of the message payload. SNMPv3 VACM provides access control to determine whether a specific type of access (read or write) to the management information is allowed.

In addition, the JUNOS SNMP agent software accepts IPv4 and IPv6 addresses for transport over IPv4 and IPv6. For IPv6, the JUNOS software supports the following IPv6 over SNMP:

- SNMP data over IPv6 networks
- IPv6-specific MIB data
- SNMP agents for IPv6

System Logging Severity Levels for SNMP Traps

For some traps, when a trap condition occurs, regardless of whether the SNMP agent sends a trap to an NMS, the trap is logged if the system logging is configured to log an event with that system logging severity level. For more information about system logging severity levels, see the *JUNOS System Basics Configuration Guide*.

For more information on system logging severity levels for standard traps, see "Standard SNMP Traps" on page 119. For more information on system logging severity levels for enterprise-specific traps, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 111.

Chapter 5
Configuring SNMP

To configure the Simple Network Management Protocol (SNMP), include the following statements at the [edit snmp] hierarchy level of the configuration:

```
snmp {
    community community-name {
        authorization authorization;
        clients {
            address restrict;
        }
        view view-name;
    }
    contact contact;
    description description;
    engine-id {
        (local engine-id | use-mac-address | use-default-ip-address);
    }
    filter-duplicates;
    interface [ interface-names ];
    location location;
    name name;
    nonvolatile {
        commit-delay seconds;
    }
    rmon {
        alarm index {
            description text-description;
            falling-event-index index;
            falling-threshold integer;
            interval seconds;
            rising-event-index index;
            falling-threshold integer;
            sample-type type;
            startup-alarm alarm;
            variable oid-variable;
        }
        event index {
            community community-name;
            description text-description;
            type type;
        }
    }
```

```
        traceoptions {
            file size size files number;
            flag flag;
        }
        trap-group group-name {
            categories [ categories ];
            destination-port <port-number>;
            targets {
                address;
            }
            version (all | v1 | v2);
        }
        trap-options {
            agent-address outgoing-interface;
            source-address address;
        }
        view view-name {
            oid object-identifier (include | exclude);
        )
    }
```

For information about configuring Remote Monitoring (RMON) alarms and events, see “Configuring RMON Alarms and Events” on page 185 and “Summary of RMON Alarm and Event Configuration Statements” on page 203.

By default, SNMP is disabled.

This chapter describes the minimum required configuration and discusses the following tasks for configuring SNMP:

Minimum SNMP Configuration

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

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

The community defined here as public grants read access to all MIB data to any client.

Configuring the System Contact

You can specify an administrative contact for each system being managed by SNMP. This name is placed into the MIB II sysContact object. To configure a contact name, include the contact statement at the [edit snmp] hierarchy level:

```
[edit snmp]
contact contact;
```

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

Example: Configuring the System Contact

Define the system contact:

```
[edit]
snmp {
    contact "Juniper Berry, (650) 555-1234";
}
```

Configuring the System Location

You can specify the location of each system being managed by SNMP. This string is placed into the MIB II sysLocation object. To configure a system location, include the location statement at the [edit snmp] hierarchy level:

```
[edit snmp]
location location;
```

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

Example: Configuring the System Location

Specify where the system is located:

```
[edit]
snmp {
    location "Row 11, Rack C";
}
```

Configuring the System Description

You can specify a description for each system being managed by SNMP. This string is placed into the MIB II sysDescription object. To configure a description, include the description statement at the [edit snmp] hierarchy level:

```
[edit snmp]
description description;
```

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

Example: Configuring the System Description

Specify the system description:

```
[edit]
snmp {
    description "M40 router with 8 FPCs";
}
```

Filtering Duplicate SNMP Requests

By default, filtering duplicate get, getNext, and getBulk SNMP requests is disabled. If a network management station (NMS) retransmits a Get, GetNext, or GetBulk SNMP request too frequently to the router, it might interfere with the processing of previous requests and slow down the response time of the agent. Filtering these duplicate requests improves the response time of the SNMP agent. The JUNOS software uses the following information to determine if an SNMP request is a duplicate:

- Source IP address of the SNMP request
- Source UDP port of the SNMP request
- Request ID of the SNMP request

To filter duplicate SNMP requests, include the filter-duplicates statement at the [edit snmp] hierarchy level:

```
[edit snmp]
filter-duplicates;
```

Configuring the Commit Delay Timer

When the router first receives an SNMP nonvolatile set request, a JUNOScript session opens and prevents other users or applications from changing the candidate configuration (equivalent to the command-line interface [CLI] configure exclusive command). If the router does not receive new SNMP set requests within 5 seconds (the default value), the candidate configuration is committed and the JUNOScript session closes (the configuration lock is released). If the router receives new SNMP set requests while the candidate configuration is being committed, the SNMP set request is rejected and an error is generated. If the router receives new SNMP set requests before 5 seconds have elapsed, the commit-delay timer (the length of time between when the last SNMP request is received and the commit is requested) resets to 5 seconds.

By default, the timer is set to 5 seconds. To configure the timer for the SNMP set reply and start of the commit, include the commit-delay statement at the [edit snmp nonvolatile] hierarchy level:

```
[edit snmp nonvolatile]
commit-delay seconds;
```

seconds is the length of the time between when the SNMP request is received and the commit is requested for the candidate configuration. For more information about the configure exclusive command and locking the configuration, see the *JUNOS System Basics Configuration Guide* and the *JUNOScript API References*.

Configuring the System Name

To specify the system name override, include the name statement at the [edit snmp] hierarchy level:

```
[edit snmp]
name name;
```

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

Example: Configuring the System Name

Specify the system name override:

```
[edit]
snmp {
    name "snmp 1";
}
```

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, 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 on the view statement, see "view" on page 151.

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 or IPv6 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 router.

NOTE: Community names must be unique. You cannot configure the same community name at the [edit snmp community] and [edit snmp v3 snmp-community *community-index*] hierarchy levels.

Examples: Configuring the SNMP Community String

Grant read-only access to all clients. With the following configuration, the system responds to SNMP Get, GetNext, and GetBulk requests that contain the community string public:

```
[edit]
snmp {
    community public {
        authorization read-only;
    }
}
```

Grant all clients read-write access to the ping MIB and jnxPingMIB. With the following configuration, the system responds to SNMP Get, GetNext, GetBulk, and Set requests that contain the community string private and specify an OID contained in the ping MIB or jnxPingMIB hierarchy:

```
[edit]
snmp {
    view ping-mib-view {
        oid pingMIB include;
        oid jnxPingMIB include;
    community private {
        authorization read-write;
        view ping-mib-view;
        }
    }
}
```

The following configuration allows read-only access to clients with IP addresses in the range 1.2.3.4/24, and denies access to systems in the range fe80::1:2:3:4/64:

```
[edit]
snmp {
    community field-service {
        authorization read-only;
        clients {
            default restrict; # Restrict access to all SNMP clients not explicitly
                              # listed on the following lines.
            1.2.3.4/24;     # Allow access by all clients in 1.2.3.4/24; except
            fe80::1:2:3:4/64 restrict;    # fe80::1:2:3:4/64
        }
    }
}
```

Configuring SNMP Trap Options and Groups

Some carriers have more than one trap receiver that forwards traps to a central NMS. This allows for more than one path for SNMP traps from a router to the central NMS through different trap receivers. A router can be configured to send the same copy of each SNMP trap to every trap receiver configured in the trap group.

The source address in the IP header of each SNMP trap packet is set to the address of the outgoing interface by default. When a trap receiver forwards the packet to the central NMS, the source address is preserved. The central NMS, looking only at the source address of each SNMP trap packet, assumes that each SNMP trap came from a different source.

In reality, the SNMP traps came from the same router, but each left the router through a different outgoing interface.

The statements discussed in the following sections are provided to allow the NMS to recognize the duplicate traps and to distinguish SNMPv1 traps based on the outgoing interface.

To configure SNMP trap options and trap groups, include the trap-options and trap-group statements at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
      address;
    }
    version (all | v1 | v2);
}
```

This section includes the following topics:

Configuring SNMP Trap Options

Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router to a single address regardless of the outgoing interface. In addition, you can set the agent address of the SNMPv1 traps. For more information on the contents of SNMPv1 traps, see RFC 1157.

NOTE: SNMP cannot be associated with any routing instances other than the master routing instance.

To configure SNMP trap options, include the trap-options statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
```

You must also configure a trap group for the trap options to take effect. For information about trap groups, see "Configuring SNMP Trap Groups" on page 35.

This section contains the following topics:

Configuring the Source Address for SNMP Traps

You can configure the source address of trap packets in two ways: lo0 or a valid IPv4 address configured on one of the router interfaces. The value lo0 indicates that the source address of the SNMP trap packets will be set to the lowest loopback address configured on the interface lo0.

To specify a valid interface address as the source address for SNMP traps on one of the router interfaces, include the source-address statement at the [edit snmp trap-options] hierarchy level:

```
[edit snmp trap-options]
source-address address;
```

address is a valid IPv4 address configured on one of the router interfaces.

To specify the source address of the SNMP traps so that they will be sent to the lowest loopback address configured on the interface lo0, include the source-address statement at the [edit snmp trap-options] hierarchy level:

```
[edit snmp trap-options]
source-address lo0;
}
```

To enable and configure the loopback address, include the address statement at the [edit interfaces lo0 unit 0 family inet] hierarchy level:

```
[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address ip-address;
        }
    }
}
```

Example: Configuring the Loopback Address as the Source Address of Trap Packets

To configure the loopback address and source address trap option:

```
[edit snmp]
trap-options {
    source-address lo0;
}
trap-group "urgent-dispatcher" {
    version v2;
    categories link startup;
    targets {
        192.168.10.22;
        172.17.1.2;
    }
}

[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.1/32;
            address 127.0.0.1/32;
        }
    }
}
```

In this example, the IP address 10.0.0.1 is the source address of every trap sent from this router.

Configuring the Agent Address for SNMP Traps

The agent address is only available in the SNMPv1 trap packets (see RFC 1157). By default, the router's default local address is used in the agent address field of the SNMPv1 trap. To configure the agent address, include the agent-address statement at the [edit snmp trap-options] hierarchy level. Currently, the agent address can only be the address of the outgoing interface:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
}
```

Example: Configuring the Outgoing Interface as the Agent Address

Configure the outgoing interface as the agent address:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
}
trap-group "urgent-dispatcher" {
    version v1;
    categories link startup;
    targets {
      192.168.10.22;
      172.17.1.2;
    }
}
```

In this example, each SNMPv1 trap packet sent has its agent address value set to the IP address of the outgoing interface.

Configuring SNMP Trap Groups

You can create and name a group of one or more types of SNMP traps and then define which systems receive the group of SNMP traps. The trap group must be configured for SNMP traps to be sent. To create an SNMP trap group, include the trap-group statement at the [edit snmp] hierarchy level:

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

Each trap group you define must have a name and one or more targets, which are the systems that receive the SNMP traps. Specify the targets by IPv4 or IPv6 address, not by hostname.

Specify the types of traps the trap group can receive in the categories statement. For information about which category traps belong to, see “Standard SNMP Traps” on page 119 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 111.

A trap group can receive the following categories:

- authentication—Authentication failures
- chassis—Chassis/environment notifications
- configuration—Configuration notifications
- link—Link-related notifications (up-down transitions, DS-3 and DS-1 line status change, IPv6 interface state change, and Passive Monitoring PIC overload)

NOTE: To send Passive Monitoring PIC overload interface traps, select the link trap category.

- remote-operations—Remote operation notifications
- rmon-alarm—Alarm for RMON events
- routing—Routing protocol notifications
- sonet-alarms—SONET/SDH alarms

NOTE: If you omit the SONET/SDH subcategories, all SONET/SDH trap alarm types are included in trap notifications.

If you include SONET/SDH subcategories, only those SONET/SDH trap alarm types are include in trap notifications.

  - loss-of-light—Loss of light alarm notification
  - pll-lock—PLL lock alarm notification
  - loss-of-frame—Loss of frame alarm notification
  - loss-of-signal—Loss of signal alarm notification
  - severely-errored-frame—Severely errored frame alarm notification
  - line-ais—Line AIS alarm notification
  - path-ais—Path AIS alarm notification
  - loss-of-pointer—Loss of pointer alarm notification
  - ber-defect—SONET/SDH bit error rate alarm defect notification
  - ber-fault—SONET/SDH error rate alarm fault notification

- line-remote-defect-indication—Line remote defect indication alarm notification
- path-remote-defect-indication—Path remote defect indication alarm notification
- remote-error-indication—Remote error indication alarm notification
- unequipped—Unequipped alarm notification
- path-mismatch—Path mismatch alarm notification
- loss-of-cell—Loss of cell delineation alarm notification
- vt-ais—VT AIS alarm notification
- vt-loss-of-pointer—VT loss of pointer alarm notification
- vt-remote-defect-indication—VT remote defect indication alarm notification
- vt-unequipped—VT unequipped alarm notification
- vt-label-mismatch—VT label mismatch error notification
- vt-loss-of-cell—VT loss of cell delineation notification

- startup—System warm and cold starts
- vrrp-events—VRRP events such as new-master or authentication failures

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 on the version statement, see version on page 151.

Example: Configuring SNMP Trap Groups

Set up a trap notification list named urgent-dispatcher for link and startup traps. This list is used to identify the network management hosts (1.2.3.4 and fe80::1:2:3:4) to which traps generated by the local router should be sent. The name specified for a trap group is used as the SNMP community string when the agent sends traps to the listed targets.

```
[edit]
snmp {
    trap-group "urgent-dispatcher" {
        version v2;
        categories link startup;
        targets {
            1.2.3.4;
            fe80::1:2:3:4;
        }
    }
}
```

Configuring the Interfaces on Which SNMP Requests Can Be Accepted

By default, all router 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 from interfaces not listed are discarded.

Example: Configuring Secured Access List Checking

Grant SNMP access privileges only to devices on interfaces so-0/0/0 and at-1/0/1. The following example does this by configuring a list of logical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0.0 so-0/0/0.1 at-1/0/1.0 at-1/0/1.1 ];
}
```

The following example grants the same access by configuring a list of physical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0 at-1/0/1 ];
}
```

Configuring MIB Views

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 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. 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-name*] hierarchy level:

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

Example: Ping Proxy MIB

Restrict the ping-mib community to read and write access of the ping MIB and jnxpingMIB only. Read or write access to any other MIB using this community is not allowed.

```
[edit snmp]
view ping-mib-view {
    oid 1.3.6.1.2.1.80 include;      #pingMIB
    oid jnxPingMIB include;          #jnxPingMIB
}
community ping-mib {
    authorization read-write;
    view ping-mib-view;
}
```

For more information on the ping MIB, see RFC 2925 and "Juniper Networks Enterprise-Specific MIBs" on page 105.

Tracing SNMP Activity

To trace SNMP activity, include the traceoptions statement at the [edit snmp] hierarchy level:

```
[edit snmp]
traceoptions {
    file size size files number;
    flag flag;
}
```

The output of the tracing operations is placed into log files in the /var/log directory. Each log file is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:

- chassisd
- craftd
- ilmid
- mib2d
- rmopd
- serviced
- snmpd

You can use the file statement to control log file generation. The size statement limits the size (in kilobytes) of each log file before it is closed and compressed and a new file opened in its place. The files statement limits the total number of log files archived for each SNMP agent.

You can specify one or more of the following values for the flag option:

- all—Trace all SNMP events
- general—Trace general events
- interface-stats—Trace physical and logical interface statistics
- nonvolatile-sets—Trace nonvolatile SNMP set request handling
- pdu—Trace SNMP request and response packets
- protocol-timeouts—Trace SNMP response timeouts
- routing-socket—Trace routing socket calls
- subagent—Trace subagent restarts
- timer—Trace internal timer events
- varbind-error—Trace variable binding errors

Example: Tracing SNMP Activity

Trace information on SNMP packets:

```
[edit]
snmp {
    traceoptions {
        file size 10k files 5;
        flag pdu;
        flag protocol-timeouts;
        flag varbind-error;
    }
}
```

Configuring the Local Engine ID

For information about configuring a local engine ID as the administratively unique identifier for an SNMPv3 engine, see “Configuring the Local Engine ID” on page 48.

Chapter 6

SNMPv3 Overview

In contrast to SNMPv1 and SNMPv2, SNMPv3 supports authentication and encryption. SNMPv3 uses the user-based security model (USM) for message security and the view-based access control model (VACM) for access control. USM specifies authentication and encryption. VACM specifies access-control rules.

USM uses the concept of a user for which security parameters (levels of security, authentication, privacy protocols, and keys) are configured for both the agent and the manager. Messages sent using USM are better protected than messages sent with community strings, where passwords are sent in the clear. With USM, messages exchanged between the manager and the agent can have data integrity checking and data origin authentication. USM protects against message delays and message replays by using time indicators and request IDs. Encryption is also available.

To complement the USM, SNMPv3 uses the VACM, a highly granular access-control model for SNMPv3 applications. Based on the concept of applying security policies to the name of the groups querying the agent, the agent decides whether the group is allowed to view or change specific Management Information Base (MIB) objects. VACM defines collections of data (called views), groups of data users, and access statements that define which views a particular group of users can use for reading, writing, or receiving traps.

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. The tag defines a set of target addresses to receive a trap. The notify filter defines access to a collection of trap OIDs. The target address defines a management application's address and other attributes to be used in sending notifications. Target parameters define the message processing and security parameters to be used in sending notifications to a particular management target.

To configure SNMPv3, perform the following tasks:

- Creating SNMPv3 Users on page 49
- Configuring MIB Views on page 54
- Defining Access Privileges for an SNMP Group on page 55
- Configuring SNMP Traps on page 62
- Configuring SNMP Informs on page 72

Chapter 7

Configuring SNMPv3

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

```
[edit snmp]
engine-id {
    (local engine-id | use-fxp0-mac-address | use-default-ip-address);
}
view view-name {
    oid object-identifier (include | exclude);
}

[edit snmp v3]
notify name {
    tag tag-name;
    type (trap | inform);
}
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;
    inform-retry-count number;
    inform-timeout seconds;
    port <port-number>;
    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-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

```
usm {
    (local-engine | remote-engine engine-id) {
        user username {
            authentication-md5 {
                authentication-password authentication-password;
            }
            authentication-none;
            authentication-sha {
                authentication-password authentication-password;
            }
            privacy-3des {
                privacy-password privacy-password;
            }
            privacy-aes128 {
                privacy-password privacy-password;
            }
            privacy-des {
                privacy-password privacy-password;
            }
            privacy-none;
        }
    }
}
vacm {
    access {
        group group-name {
            default-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;
            }
        }
    }
}
```

This section includes the following topics for configuring SNMPv3:

Minimum SNMPV3 Configuration

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

```
[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-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

```
usm {
local-engine {
        user username {
        }
    }
}
vacm {
    access {
      group group-name {
        default-context-prefix {
          security-model (any | usm | v1 | v2c) {
            security-level (authentication | none | privacy) {
            }
          }
        }
      }
    }
    security-to-group {
      security-model (usm | v1 | v2c) {
        security-name security-name {
          group group-name;
        }
      }
    }
}
```

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

Configuring the Local Engine ID

By default, the local engine ID uses the default IP address of the router. The local engine ID is the administratively unique identifier for the SNMPv3 engine. This statement is optional. To configure the local engine ID, include the engine-id statement at the [edit snmp] hierarchy level:

```
[edit snmp]
engine-id {
    (local engine-id-suffix | use-default-ip-address | use-mac-address);
}
```

- local *engine-id-suffix*—The engine ID suffix is explicitly configured.
- use-default-ip-address—The engine ID suffix is generated from the default IP address.
- use-mac-address—The SNMP engine identifier is generated from the Media Access Control (MAC) address of the management interface on the routing platform.

The local engine ID is defined as the administratively unique identifier of an SNMPv3 engine, and is used for identification, not for addressing. There are two parts of an engine ID: prefix and suffix. The prefix is formatted according to the specifications defined in RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.* You can configure the suffix here.

NOTE: SNMPv3 authentication and encryption keys are generated based on the associated passwords and the engine ID. If you configure or change the engine ID, you must commit the new engine ID before you configure SNMPv3 users. Otherwise the keys generated from the configured passwords will be based on the previous engine ID.

For the engine ID, we recommend using the MAC address of fxp0.

Creating SNMPv3 Users

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

NOTE: You can only configure 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;
```

This section discusses the following topics:

- Configuring the Authentication Type on page 50
- Configuring the Encryption Type on page 51
- Example: Creating SNMPv3 Users Configuration on page 53

Configuring the Authentication Type

By default, the authentication type is set to none.

This section includes the following topics:

- Configuring the MD5 Authentication on page 50
- Configuring the SHA Authentication on page 51
- Configuring No Authentication on page 51

Configuring the MD5 Authentication

To configure the message digest algorithm (MD5) as the authentication type for an SNMPv3 user, include the authentication-md5 statement 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-password is the password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Configuring the SHA Authentication

To configure the secure hash algorithm (SHA) as the authentication type for an SNMPv3 user, include the authentication-sha statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[edit snmp v3 usm local-engine user username]
authentication-sha {
    authentication-password authentication-password;
}
```

authentication-password is the password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Configuring No Authentication

To configure no authentication for an SNMPv3 user, include the authentication-none statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[edit snmp v3 usm local-engine user username]
authentication-none;
```

Configuring the Encryption Type

By default, encryption is set to none.

NOTE: Before you configure encryption, you must configure the MD5 or SHA authentication.

Before you configure the privacy-3des and privacy-aes128 statements, you must install the jcrypto package.

This section includes the following topics:

Configuring the Advanced Encryption Standard Algorithm

To configure the Advanced Encryption Standard (AES) algorithm for an SNMPv3 user, include the privacy-aes128 statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[snmp v3 usm local-engine user username]
privacy-aes128 {
    privacy-password privacy-password;
}
```

privacy-password is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Configuring the Data Encryption Algorithm

To configure the data encryption algorithm (DES) for an SNMPv3 user, include the privacy-des statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[edit snmp v3 usm local-engine user username]
privacy-des {
    privacy-password privacy-password;
}
```

privacy-password is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Configuring Triple DES

To configure triple DES for an SNMPv3 user, include the privacy-3des statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[snmp v3 usm local-engine user username]
privacy-3des {
    privacy-password privacy-password;
}
```

privacy-password is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Configuring No Encryption

To configure no encryption for an SNMPv3 user, include the privacy-none statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[edit snmp v3 usm local-engine user username]
privacy-none;
```

Example: Creating SNMPv3 Users Configuration

Define SNMPv3 users:

```
[edit]
snmp {
    v3 {
        usm {
            local-engine {
                user user1 {
                  authentication-md5 {
                     authentication-password authentication-password;
                   }
                  privacy-des {
                    privacy-password password;
                  }
                }
                user user2 {
                  authentication-sha {
                      authentication-password authentication-password;
                 }
                 privacy-none;
                }
                user user3 {
                  authentication-none;
                  privacy-none;
                }
                user user4 {
                  authentication-md5 {
                      authentication-password authentication-password;
                  }
                  privacy-none {
                     privacy-password privacy-password;
                  }
                }
```

```
            user user5 {
              authentication-sha {
                  authentication-password authentication-password;
              }
              privacy-aes128 {
                 privacy-password privacy-password;
              }
            }
          }
        }
      }
    }
}
```

Configuring MIB Views

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. For SNMPv3, you must associate the view with a group name configured at the [edit snmp v3 vacm] hierarchy level.

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 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. To enable a view, you must associate the view with a community. To enable a view for SNMPv3, you must associate the view with a group name configured at the [edit snmp v3 vacm] hierarchy level.

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-name*] hierarchy level:

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

For information about how to associate MIB views to an SNMPv3 user group, see “Associating MIB Views with an SNMP User Group” on page 58.

Example: Ping Proxy MIB

Restrict the ping MIB community to read and write access of the ping MIB and jnxpingMIB only. Read or write access to any other MIB using this community is not allowed.

```
[edit snmp]
view ping-mib-view {
    oid 1.3.6.1.2.1.80 include;      #pingMIB
    oid jnxPingMIB include;          #jnxPingMIB
}
community ping-mib {
    authorization read-write;
    view ping-mib-view;
}
```

For more information on the ping MIB, see RFC 2925 and "Juniper Networks Enterprise-Specific MIBs" on page 105.

Defining Access Privileges for an SNMP Group

SNMPv3 uses the view-based access control model (VACM), which allows you to configure the access privileges granted to a group. Access is controlled by filtering the MIB objects available for a specific operation through a predefined view. You assign views to determine the objects that are visible for read, write, and notify operations for a particular group, using a particular context (only the default context is supported), a particular security model (v1,v2c, or usm), and particular security level (authenticated, privacy, or none). For information about how to configure views, see "Configuring MIB Views" on page 54.

You define user access to management information at the [edit snmp v3 vacm] hierarchy level. All access control within VACM operates on groups, which are collections of users as defined by USM, or community strings as defined in the SNMPv1 and SNMPv2c security models. The term *security-name* refers to these generic end users. The group to which a specific security name belongs is configured at the [edit snmp v3 vacm security-to-group] hierarchy level. That security name can be associated with a group defined at the [edit snmp v3 vacm security-to-group] hierarchy level. A group identifies a collection of SNMP users that share the same access policy. You then define the access privileges associated with a group at the [edit snmp v3 vacm access] hierarchy level. Access privileges are defined using views. For each group, you can apply different views depending on the SNMP operation; for example, reads (get, getNext, or getbulk) writes (set), notifications, the security level used (authentication, privacy, or none), and the security model (v1, v2c, or usm) used within an SNMP request.

You configure members of a group with the security-name statement. For v3 packets using USM, the security name is the same as the username. For SNMPv1 or SNMPv2c packets, the security name is determined based on the community string. Security names are specific to a security model. If you are also configuring VACM access policies for SNMPv1 or SNMPv2c packets, you must assign security names to groups for each security model (SNMPv1 or SNMPv2c) at the [edit snmp v3 vacm security-to-group] hierarchy level. You must also associate a security name with an SNMP community at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

To configure the access privileges for an SNMP group, include statements at the [edit snmp v3 vacm] hierarchy level:

```
[edit snmp v3 vacm]
access {
    group group-name {
        default-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;
            }
        }
    }
}
```

This section describes the following topics related to defining privileges for an SNMP group:

- Configuring the Access Privileges Granted to a Group on page 56
- Assigning Security Names to Groups on page 60

Configuring the Access Privileges Granted to a Group

This section includes the following topics:

- Configuring the Group on page 57
- Configuring the Security Model on page 57
- Configuring the Security Level on page 57
- Associating MIB Views with an SNMP User Group on page 58
- Example: Access Privilege Configuration on page 59

Configuring the Group

To configure the access privileges granted to a group, include the group statement at the [edit snmp v3 vacm access] hierarchy level:

```
[edit snmp v3 vacm access]
group group-name;
```

group-name is a collection of SNMP users that belong to a common SNMP list that defines an access policy. Users belonging to a particular SNMP group inherit all access privileges granted to that group.

Configuring the Security Model

To configure the security model, include the security-model statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix]
security-model (any | usm | v1 | v2c);
```

- any—Any security model
- usm—SNMPv3 security model
- v1—SNMPV1 security model
- v2c—SNMPv2c security model

Configuring the Security Level

To configure the access privileges granted to packets with a particular security level, include the security-level statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c)] hierarchy level:

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

- none—Provides no authentication and no encryption.
- authentication—Provides authentication but no encryption.
- privacy—Provides authentication and encryption.

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 SNMPv2c 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.

Associating MIB Views with an SNMP User Group

MIB views define access privileges for members of a group. Separate views can be applied for each SNMP operation (read, write, and notify) within each security model (usm, v1, and v2c) and each security level (authentication, none, and privacy) supported by SNMP.

To associate MIB views with an SNMP user group, include the following statements at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-mode (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-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;
```

NOTE: You must associate at least one view (notify, read, or write) at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level.

You must configure the MIB view at the [edit snmp view *view-name*] hierarchy level. For information about how to configure MIB views, see "Configuring MIB Views" on page 54.

This section describes the following topics related to this configuration:

- Configuring the Notify View on page 58
- Configuring the Read View on page 59
- Configuring the Write View on page 59

Configuring the Notify View

To associate notify access with an SNMP user group, include the notify-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

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

view-name specifies the notify access, which is a list of notifications that can be sent to each user in an SNMP group. A view name cannot exceed 32 characters.

Configuring the Read View

To associate a read view with an SNMP group, include the read-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

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

view-name specifies read access for an SNMP user group. A view name cannot exceed 32 characters.

Configuring the Write View

To associate a write view with an SNMP user group, include the write-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

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

view-name specifies write access for an SNMP user group. A view name cannot exceed 32 characters.

Example: Access Privilege Configuration

Define access privileges:

```
[edit snmp v3]
access {
    group group1 {
        default-context-prefix {
            security-model usm {       #Define an SNMPv3 security model
                security-level privacy {
                  notify-view nv1;
                  read-view rv1;
                  write-view wv1;
                }
            }
        }
    }
    group group2 {
        default-context-prefix {
            security-model usm {       #Define an SNMPv3 security model
                security-level authentication {
                    read-view rv2;
                    write-view wv2;
                }
            }
        }
    }
```

```
        group group3 {
            default-context-prefix {
                security-model v1 {          #Define an SNMPv1 security model
                    security-level none {
                        read-view rv3;
                        write-view wv3;
                    }
                }
            }
        }
    }
}
```

Assigning Security Names to Groups

To assign security names to groups, include the following statements at the [edit snmp v3 vacm security-to-group] hierarchy level:

```
[edit snmp v3 vacm security-to-group]
security-model (usm | v1 | v2c) {
    security-name security-name {
        group group-name;
    }
}
```

This section includes the following topics:

- Configuring the Security Model on page 60
- Configuring the Security Name on page 61
- Configuring the Group on page 61
- Example: Security Group Configuration on page 62

Configuring the Security Model

To configure the security model, include the security-model statement at the [edit snmp v3 vacm security-to-group] hierarchy level:

```
[edit snmp v3 vacm security-to-group]
security-model (usm | v1 | v2c);
```

- usm—SNMPv3 security model
- v1—SNMPv1 security model
- v2c—SNMPv2 security model

Configuring the Security Name

To associate a security name with a user or community string, include the security-name statement at the [edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c)] hierarchy level:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c)]
security-name security-name;
```

security-name is the username configured at the [edit snmp v3 usm local-engine user *username*] hierarchy level. For SNMPv1 and SNMPv2c, the security name is the community string configured at the [edit snmp v3 snmp-community *community-index*] hierarchy level. For information about configuring usernames, see "Creating SNMPv3 Users" on page 49. For information about configuring a community string, see "Configuring the SNMP Community" on page 77.

NOTE: The USM security name is separate from the SNMPv1 and SNMPv2c security name. If you are supporting SNMPv1 and SNMPv2c, you must configure separate security names within the security-to-group configuration at the [edit snmp v3 vacm access] hierarchy level.

Configuring the Group

After you have created users, v1, or v2 security names, you associate them with a group. A group is a set of security names belonging to a particular security model. A group defines the access rights for all users belonging to it. Access rights define what SNMP objects can be read, written to, or created. A group also defines what notifications a user is allowed to receive.

If you already have a group that is configured with all of the view and access permissions that you want to give a user, you can add the user to that group. If you want to give a user view and access permissions that no other groups have, or if you do not have any groups configured, create a group and add the user to it.

To configure the access privileges granted to a group, include the group statement at the [edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c) security-name *security-name*] hierarchy level:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c) security-name
    security-name]
group group-name;
```

group-name identifies a collection of SNMP security names that share the same access policy. For more information about groups, see "Defining Access Privileges for an SNMP Group" on page 55.

Example: Security Group Configuration

Assign security names to groups:

```
vacm {
    security-to-group {
        security-model usm {
            security-name user1 {
                group group1;
            }
            security-name user2 {
                group group2;
            }
            security-name user3 {
                group group3;
            }
        }
    }
```

Configuring SNMP Traps

In SNMPv3, traps and informs are created by configuring the notify, target-address, and target-parameters parameters. Traps are unconfirmed notifications and informs are confirmed notifications. This section describes how to configure SNMP traps. For information on configuring SNMP informs, see “Configuring SNMP Informs” on page 72.

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 | inform);
}
notify-filter name {
    oid object-identifier (include | exclude);
}
target-address target-address-name {
    address address;
    address-mask address-mask;
    port <port-number>;
    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-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

This section includes the following topics:

Configuring the Trap Notification

The notify statement specifies the type of notification (trap) and contains a single tag. The tag defines a set of target addresses to receive a trap. The tag list contains one or more tags and is configured at the [edit snmp v3 target-address *target-address-name*] hierarchy level. If the tag list contains this tag, the JUNOS software sends a notification to all the target addresses associated with this tag.

To configure the trap notifications, include the notify statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
notify name {
    tag tag-name;
    type trap;
}
```

name is the name assigned to the notification.

tag-name defines the target addresses that are sent this notification. All the target-addresses that have this tag in their tag list are sent this notification. The *tag-name* is not included in the notification.

trap is the type of notification.

NOTE: Each notify entry name must be unique.

The JUNOS software supports two types of notification: trap and inform.

For information about how to configure the tag list, see "Configuring the Tag List" on page 66.

Example: Trap Notification Configuration

Specify three sets of destinations to send traps:

```
[edit snmp v3]
notify n1 {
    tag router1;
    type trap;
}
notify n2 {
    tag router2;
    type trap
}
notify n3 {
    tag router3;
    type trap;
}
```

Configuring the Trap Notification Filter

SNMPv3 uses the notify filter to define which traps (or which objects from which traps) will be sent to the network management station (NMS). The trap notification filter limits the type of traps that are sent to the NMS.

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).

To configure the trap notifications filter, include the notify-filter statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
notify-filter profile-name;
```

profile-name is the name assigned to the notify filter.

By default, the OID is set to include. To define access to traps (or objects from traps), include the oid statement at the [edit snmp v3 notify-filter *profile-name*] hierarchy level:

```
[edit snmp v3 notify-filter profile-name]
oid oid (include | exclude);
```

oid is the object identifier. All MIB objects represented by this statement have the specified OID as a prefix. It can be specified either by a sequence of dotted integers or by a subtree name.

- include—Include the subtree of MIB objects represented by the specified OID.
- exclude—Exclude the subtree of MIB objects represented by the specified OID.

Configuring the Trap Target Address

The target address defines a management application's address and parameters that are used in sending notifications. It can also identify management stations that are allowed to use specific community strings. When you receive a packet with a recognized community string and a tag is associated with it, the JUNOS software looks up all the target addresses with this tag and verifies that the source address of this packet matches one of the configured target addresses.

NOTE: You must configure the address mask when you configure the SNMP community.

To specify where you want the traps to be sent and define what SNMPv1 and SNMP2vc packets are allowed, include the target-address statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
target-address target-address-name;
```

target-address-name is the string that identifies the target address.

To configure the target address properties, include the following statements at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
address address;
address-mask address-mask;
port <port-number>;
tag-list [ tag-list ];
target-parameters target-parameters-name;
```

This section includes the following topics:

Configuring the Address

To configure the address, include the address statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
address address;
```

address is the SNMP target address.

Configuring the Address Mask

The address mask specifies a set of addresses that are allowed to use a community string and verifies the source addresses for a group of target addresses.

To configure the address mask, include the address-mask statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
address-mask address-mask;
```

address-mask combined with the address define a range of addresses. For information about how to configure the community string, see "Configuring the SNMP Community" on page 77.

Configuring the Port

By default, the UDP port is set to 162. To configure the port, include the port statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
port <port-number>;
```

port is the SNMP target port number.

Configuring the Tag List

Each target-address statement can have one or more tags configured in its tag list. Each tag can appear in more than one tag list. When a significant event occurs on the network device, the tag list identifies the targets to which a notification is sent.

To configure the tag list, include the tag-list statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
tag-list [ tag-list ];
```

tag-list specifies one or more tags.

For information about how to specify a tag at the [edit snmp v3 notify *notify-name*] hierarchy level, see "Configuring the Trap Notification" on page 63.

Example: Configuring the Tag List

In the following example, two tag entries (router1 and router2) are defined at the [edit snmp v3 notify *notify-name*] hierarchy level. When an event triggers a notification, the JUNOS software sends a trap to all target addresses that have router1 or router2 configured in their target-address tag list. This results in the first two targets getting one trap each, and the third target getting two traps.

```
[edit snmp v3]
notify n1 {
   tag router1;            # Identifies a set of target addresses
   type trap;              # Defines the type of notification
}
notify n2 {
   tag router2;
   type trap;
}
target-address ta1 {
   address 10.1.1.1
   address-mask 255.255.255.0;
   port 162;
   tag-list router1;
   target-parameters tp1;
}
target-address ta2 {
   address 10.1.1.2
   address-mask 255.255.255.0;
   port 162;
   tag-list router2;
   target-parameters tp2;
}
target-address ta3 {
   address 10.1.1.3
   address-mask 255.255.255.0;
   port 162;
   tag-list [router1 router2];  #Define multiple tags in the target address tag
   target-parameters tp3;     #list
}
```

NOTE: When you configure SNMP traps, make sure your configured access privileges allow the traps to be sent. Configure access privileges at the [edit snmp v3 vacm access] hierarchy level.

Applying Target Parameters

The target-parameters statement at the [edit snmp v3] hierarchy level applies the target parameters configured at the [edit snmp v3 target-parameters *target-parameters-name*] hierarchy level.

To reference configured target parameters, include the target-parameters statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
target-parameters target-parameters-name;
```

target-parameters-name is the name associated with the message processing and security parameters that are used in sending notifications to a particular management target.

Defining the Trap Target Parameters

Target parameters define the message processing and security parameters that are used in sending notifications to a particular management target.

To define a set of target parameters, include the target-parameters statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
target-parameters target-parameters-name;
```

target-parameters-name is the name assigned to the target parameters.

To configure target parameter properties, include the following statements at the [edit snmp v3 target-parameters *target-parameter-name*] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-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;
}
```

This section includes the following topics:

Applying the Trap Notification Filter

To apply the trap notification filter, include the notify-filter statement at the [edit snmp v3 target-parameters *target-parameter-name*] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name]
notify-filter profile-name;
```

profile-name is the name of a configured notify filter. For information about configuring notify filters, see "Configuring the Trap Notification Filter" on page 64.

Configuring the Target Parameters

To configure target parameter properties, include the following statements at the [edit snmp v3 target-parameters *target-parameter-name* parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
message-processing-model (v1 | v2c | v3);
security-model (usm | v1 | v2c);
security-level (authentication | none | privacy);
security-name security-name;
```

This section includes the following topics:

Configuring the Message Processing Model

The Message Processing Model defines which version of SNMP to use when generating SNMP notifications. To configure the message processing model, include the message-processing statement at the [edit snmp v3 target-parameters *target-parameter-name* parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
message-processing-model (v1 | v2c | v3);
```

- v1—SNMPv1 message processing model
- v2c—SNMPv2c message processing model
- v3—SNMPV3 message processing model

Configuring the Security Model

To define the security model to use when generating SNMP notifications, include the security-model statement at the [edit snmp v3 target-parameters *target-parameter-name* parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-model (usm | v1 | v2c);
```

- usm—SNMPv3 security model
- v1—SNMPv1 security model
- v2c—SNMPv2c security model

Configuring the Security Level

The security-level statement specifies whether the trap is authenticated and encrypted before it is sent.

To configure the security level to use when generating SNMP notifications, include the security-level statement at the [edit snmp v3 target-parameters *target-parameter-name* parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-level (authentication | none | privacy);
```

- authentication—Provides authentication but no encryption.
- none—No security. Provides no authentication and no encryption.
- privacy—Provides authentication and encryption.

NOTE: If you are configuring the SNMPv1 or SNMPV2c security model, use none as your security level. If you are configuring the SNMPv3 (USM) security model, use the authentication or privacy security level.

Configuring the Security Name

To configure the security name to use when generating SNMP notifications, include the security-name statement at the [edit snmp v3 target-parameters *target-parameter-name* parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-name security-name;
```

If the USM security model is used, the security-name identifies the user that is used when generating the notification. If the v1 or v2c security models are used, security-name identifies the SNMP community used when generating the notification.

NOTE: The access privileges for the group associated with a security name must allow this notification to be sent.

If you are using the v1 or v2 security models, the security name at the [edit snmp v3 vacm security-to-group] hierarchy level must match the security name at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

Example: Trap Configuration

Define traps:

```
[edit snmp v3]
notify n1 {
     tag router2;                # Identifies the target address
     type trap;                  # Defines the type of notification
}
notify-filter nf1 {
     oid .1 include;             # Filters the type of traps that are sent to the NMS
}
target-address ta1 {             # Includes multiple addresses
     address 10.1.1.1;
     address-mask 255.255.255.0;
     port 162;
     tag-list router2;
     target-parameters tp1;      # Applies configured target parameters
}
target-parameters tp1 {          # Defines target parameters
     notify-filter nf1;
     parameters {
          message-processing-model v1;
          security-model v1‘;
          security-level none;
          security-name john;
      }
}
```

Configuring SNMP Informs

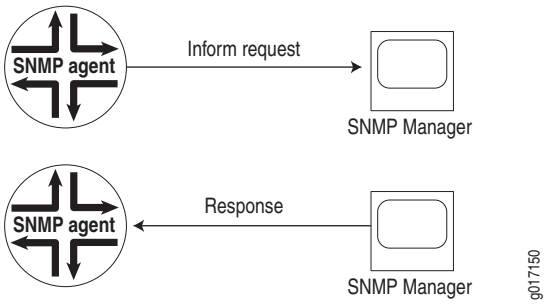
JUNOS software supports two types of notifications: traps and informs. With traps, the receiver does not send any acknowledgment when it receives a trap. Therefore, the sender cannot determine if the trap was received. A trap may be lost because a problem occurred during transmission. To increase reliability, an inform is similar to a trap except that the inform is stored and retransmitted at regular intervals until:

- 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. 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 and router resources. 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, while 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 memory, use traps.

Figure 1: Inform Request and Response



This section describes how to configure SNMP informs and includes the following topics:

- Configuring the Remote Engine and Remote User on page 73.
- Configuring the Inform Notification Type and Target Address on page 74.

For information on configuring SNMP traps, see “Configuring SNMP Traps” on page 62.

Configuring the Remote Engine and Remote User

To send inform messages to an SNMPv3 user on a remote device, you must first specify the engine identifier for the SNMP agent on the remote device where the user resides. The remote engine ID is used to compute the security digest for authenticating and encrypting packets sent to a user on the remote host. When sending an inform message, the agent uses the credentials of the user configured on the remote engine (inform target).

To configure a remote engine and remote user to receive and respond to SNMP informs, include the following statements at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
usm {
    remote-engine engine-id {
        user username {
            authentication-md5 {
                authentication-key key;
            }
            authentication-none;
            authentication-sha{
                authentication-key key;
            }

            privacy-3des {
                privacy-key key;
            }
            privacy-aes128 {
                privacy-key key;
            }
            privacy-des {
                privacy-key key;
            }
            privacy-none;
        }
    }
}
```

For informs, remote-engine *engine-id* is the identifier for the SNMP agent on the remote device where the user resides.

For informs, user *username* is the user on a remote SNMP engine who receives the informs.

Informs generated can be unauthenticated, authenticated, or authenticated_and_encrypted, depending on the security level of the SNMPv3 user configured on the remote engine (the inform receiver). The authentication key is used for generating message authentication code (MAC). The privacy key is used to encrypt the inform PDU part of the message.

Example: Configuring the Remote Engine ID and Remote Users

The following example configures user u10 located on remote engine 0x800007E5804089071BC6D10A41 and the user's authentication and privacy keys. The keys are autogenerated from the passwords entered by the command-line interface (CLI) user.

```
[edit snmp v3]
usm {
    remote-engine 800007E5804089071BC6D10A41 {
        user u10 {
            authentication-md5 {
                authentication-key "$9$D0jP536901Riktu1IcSwY2gUj5QF3
                    /CYgQF/Cu0xN-bwgZGiqP5iH.5TF/9WLX7wYoaUkqfoaAp
                    0BEhSreW87s24aUjsY4ZDjq.RhcyWLNdbg4Zs
                    YJDHkTQ69Apu1EcyrvWQF/tuOREYg4ajHmPQF39
                    Ygz3n6At8XxNYgik.PTz7-ikmfn6vW8XVw";
                }
            privacy-des {
                privacy-key "$9$MZZXxdwYgJUjIKJGiH5T69Au0IrlM7NbeK24
                    aJDjO1IRylM8Xbwg1R24aJDjHqm5n/Ap0ORhn6evLXbwmf5T
                    /CRhSyKM5QEcleW87-Vbs4JGD.mT-VwgaZkqfTznAphSrlM8yr
                    Wx7dsYTzF36AtuO1EcpuNdwYoa69CuRhcyleM8rlaZGjq.O1IEhr";
            }
        }
    }
}
```

Configuring the Inform Notification Type and Target Address

To configure the inform notification type and target information, include the following statements at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
notify name {
    tag tag-name;
    type (trap | inform);
}
target-address target-address-name {
    address address;
    address-mask address-mask;
    inform-retry-count number;
    inform-timeout seconds;
    port <port-number>;
    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-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

notify *name* is the name assigned to the notification. Each notify entry name must be unique.

tag *tag-name* defines the target addresses that are sent this notification. The notification is sent to all target addresses that have this tag in their tag list. The *tag-name* is not included in the notification. For information about how to configure the tag list, see “Configuring the Tag List” on page 66.

type inform is the type of notification.

target-address *target-address-name* identifies the target address. The target address defines a management application's address and parameters that are used to respond to informs.

inform-timeout *seconds* is the number of seconds to wait for an acknowledgement. If no acknowledgement is received within the timeout period, the inform is retransmitted. The default timeout is 15 seconds.

inform-retry-count *number* is the maximum number of times an inform is transmitted if no acknowledgement is received. The default is 3. If no acknowledgement is received after the inform is transmitted the maximum number of times, the inform message is discarded.

message-processing-model defines which version of SNMP to use when generating SNMP notifications. Informs require a v3 message processing model.

security-model defines the security model to use when generating SNMP notifications. Informs require a usm security model.

security-level specifies whether the inform is authenticated and encrypted before it is sent. For the usm security model, the security level must be be one of the following:

- authentication—Provides authentication but no encryption.
- privacy—Provides authentication and encryption.

security-name identifies the username that is used when generating the inform.

Example: Configuring the Inform Notification Type and Target Address

In the following example, target 172.17.20.184 is configured to respond to informs. The inform timeout is 30 seconds and the maximum retransmit count is 3. The inform is sent to all targets in the tl1 list. The security model for the remote user is usm and the remote engine username is u10.

```
[edit snmp v3]
notify n1 {
    type inform;
    tag tl1;
}
notify-filter nf1 {
    oid .1.3 include;
}
target-address ta1 {
    address 172.17.20.184;
    inform-timeout 30;
    inform-retry-count 3;
    tag-list tl1;
    address-mask 255.255.255.0;
    target-parameters tp1;
}
target-parameters tp1 {
    parameters {
        message-processing-model v3;
        security-model usm;
        security-level privacy;
        security-name u10;
    }
    notify-filter nf1;
}
```

Configuring the SNMP Community

The SNMP community defines the relationship between an SNMP server system and the client systems. This statement is optional.

To configure the SNMP community, include the snmp-community statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
snmp-community community-index;
```

community-index is the index for the SNMP community.

To configure the SNMP community properties, include the following statements at the [edit snmp v3 snmp-community *community-index*] hierarchy level:

```
[edit snmp v3 snmp-community community-index]
community-name community-name;
security-name security-name;
tag tag-name;
```

This section includes the following topics:

Configuring the Community Name

The community name defines the SNMP community. The SNMP community authorizes SNMPv1 or SNMPv2c clients. The access privileges associated with the configured security name define which MIB objects are available and the operations (read, write, or notify) allowed on those objects.

To configure the SNMP community name, include the community-name statement at the [edit snmp v3 snmp-community *community-index*] hierarchy level:

```
[edit snmp v3 snmp-community community-index]
community-name community-name;
```

community-name is the community string for an SNMPv1 or SNMPv2c community. If unconfigured, it is the same as the community index.

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

NOTE: Community names must be unique. You cannot configure the same community name at the [edit snmp community] and [edit snmp v3 snmp-community *community-index*] hierarchy levels.

The configured community name at the [edit snmp v3 snmp-community *community-index*] hierarchy level is encrypted. You cannot view the community name after you have configured it and committed your changes. In the CLI, the community name is concealed.

Configuring the Security Names

To assign a community string to a security name, include the security-name statement at the [edit snmp v3 snmp-community *community-index*] hierarchy level:

```
[edit snmp v3 snmp-community community-index]
security-name security-name;
```

security-name is used when performing access control. The security-to-group configuration at the [edit snmp v3 vacm] hierarchy level identifies the group.

NOTE: This security name must match the security name configured at the [edit snmp v3 target-parameters *target-parameters-name* parameters] hierarchy level when configuring traps.

Configuring the Tag

To configure the tag, include the tag statement at the [edit snmp v3 snmp-community *community-index*] hierarchy level:

```
[edit snmp v3 snmp-community community-index]
tag tag-name;
```

tag-name identifies the address of managers that are allowed to use a community string.

Example: SNMP Community Configuration

Define an SNMP community:

```
[edit snmp v3]
snmp-community index1 {
    community-name "$9$JOZi.QF/AtOz3"; # SECRET-DATA
    security-name john;
    tag router1;                    # Identifies managers that are allowed to use
}                                   # a community string
target-address ta1 {
    address 10.1.1.1;
    address-mask 255.255.255.0; # Defines the range of addresses
    port 162;
    tag-list router1;
    target-parameters tp1;          # Apply configured target parameters
}
```

Example: SNMPv3 Configuration

Define an SNMPv3 configuration:

```
[edit snmp]
engine-id {
    use-fxp0-mac-address;
}
view jnxAlarms {
    oid 1.3.6.1.4.1.2636.3.4 include;
)
view interfaces {
    oid 1.3.6.1.2.1.2 include;
)
view ping-mib {
    oid 1.3.6.1.2.1.80 include;
)
[edit snmp v3]
notify n1 {
    tag router1;                # Identifies a set of target addresses
    type trap;                  # Defines type of notification
}
notify n2 {
    tag host1;
    type trap;
}
notify-filter nf1 {
   oid .1 include;              # Defines which traps (or which objects for which
}                               # that will be sent. In this case, include all traps.
notify-filter nf2 {
   oid 1.3.6.1.4.1 include;     # Send enterprise-specific traps only
}
notify-filter nf3 {
   oid 1.3.6.1.2.1.1.5 include; # Send BGP traps only
}
```

```
snmp-community index1 {
   community-name "$9$JOZi.QF/AtOz3";  # SECRET-DATA
   security-name john;     # Matches the security name at the target parameters
     tag host1;            # Finds the addresses that are allow to be used with
}                          # this community string
target-address ta1 {       # Associates the target address with the group
san-francisco;
     address 10.1.1.1;
     address-mask 255.255.255.0;  # Defines the range of addresses
     port 162;
     tag-list router1;
     target-parameters tp1;            # Apply configured target parameters

target-address ta2 {
     address 10.1.1.2;
     address-mask 255.255.255.0;
     port 162;
     tag-list host1;
     target-parameters tp2;
}
target-address ta3 {
     address 10.1.1.3;
     address-mask 255.255.255.0;
     port 162;
     tag-list [router1 host1];
     target-parameters tp3;
}
target-parameters tp1 {        # Define the target parameters
     notify-filter nf1;        # Specify which notify filter to apply
     parameters {
        message-processing-model v1;
        security-model v1‘;
        security-level none;
        security-name john;    # Matches the security name configured at the
     }                         # [edit snmp v3 snmp-community community-index]
}                              #hierarchy level
target-parameters tp2 {
     notify-filter nf2;
   parameters {
       message-processing-model v1;
       security-model v1‘;
       security-level none;
       security-name john;
     }
}
target-parameters tp3 {
     notify-filter nf3;
   parameters {
       message-processing-model v1;
       security-model v1‘;
       security-level none;
       security-name john;
     }
}
```

```
usm {
    local-engine {          #Define authentication and encryption for SNMP3 users.
      user user1 {
        authentication-md5 {
          authentication-password authentication-password;
        }
        privacy-des {
          privacy-password privacy-password;
          }
        }
      user user2 {
        authentication-sha {
          authentication-password authentication-password;
        }
        privacy-none;
        }
      user user3 {
        authentication-none;
        privacy-none;
      }

      user user4 {
        authentication-sha {
          authentication-password authentication-password;
        }
        privacy-aes128 {
          privacy-password privacy-password;
        }
      }
      user user5 {
        authentication-sha {
          authentication-password authentication-password;
        }
        privacy-none {
          privacy-password privacy-password;
        }
      }
    }
}
vacm {
     access {
       group san-francisco {          #Defines the access privileges for the group
          default-context-prefix {    #san-francisco
            security-model v1 {
              security-level none {
                notify-view ping-mib;
                read-view interfaces;
                write-view jnxAlarms;
              }
            }
          }
        }
    }
```

```
    security-to-group {
        security-model v1 {
            security-name john {              #Assigns john to the security group
                group san-francisco;          #san-francisco
            }
            security-name bob {
                group new-york;
            }
                security-name elizabeth {
                group chicago;
            }
        }
    }
}
```

Chapter 8

SNMP Remote Operations

A Simple Network Management Protocol (SNMP) remote operation is any process on the router that can be controlled remotely using SNMP. The JUNOS software currently provides support for two SNMP remote operations: the ping Management Information Base (MIB) and traceroute MIB, defined in RFC 2925. Using these MIBs, an SNMP client in the network management system (NMS) can:

- Start a series of operations on a router
- Receive notification when the operations are complete
- Gather the results of each operation

The JUNOS software also provides extended functionality to these MIBs in the Juniper Networks enterprise-specific extensions jnxPingMIB and jnxTraceRouteMIB. For more information about jnxPingMIB and jnxTraceRouteMIB, see "Juniper Networks Enterprise-Specific MIBs" on page 105.

This chapter covers the following topics:

SNMP Remote Operation Requirements

To use SNMP remote operations, you should be experienced with SNMP conventions. You must also configure the JUNOS software to allow the use of the remote operation MIBs.

To configure the JUNOS software for remote operations, complete the following tasks:

- Setting SNMP Views on page 84
- Setting Trap Notification for Remote Operations on page 86
- Using Variable-Length String Indexes on page 86
- Enabling Logging on page 87

Setting SNMP Views

All remote operation MIBs supported by the JUNOS software require that the SNMP clients have read-write privileges. The default SNMP configuration of the JUNOS software does not provide clients with a community string with such privileges.

To set read-write privileges for an SNMP community string, include the following statements at the [edit snmp] hierarchy level:

```
snmp {
    view view-name;
        oid object-identifier (include | exclude);
    }
    community community-name {
        authorization authorization;
        view view-name;
    }
}
```

Example: Setting SNMP Views

To create a community named remote-community that grants SNMP clients read-write access to the ping MIB, jnxPing MIB, traceroute MIB, and jnxTraceRoute MIB, include the following statements at the [edit snmp] hierarchy level:

```
snmp {
    view remote-view {
        oid 1.3.6.1.2.1.80 include;             # pingMIB
        oid 1.3.6.1.4.1.2636.3.7 include;       # jnxPingMIB
        oid 1.3.6.1.2.1.81 include;             # traceRouteMIB
        oid 1.3.6.1.4.1.2636.3.8 include;       # jnxTraceRouteMIB
    }
    community remote-community {
        view remote-view;
        authorization read-write;
    }
}
```

For more information on the community statement, see "Configuring the SNMP Community String" on page 30 and "community" on page 142.

For more information on the view statement, see "Configuring MIB Views" on page 39 and "view" on page 151.

Setting Trap Notification for Remote Operations

In addition to configuring the remote operations MIB for trap notification, you must also configure the JUNOS software. You must specify a target host for remote operations traps.

To configure trap notification for SNMP remote operations, include the categories and targets statements at the [edit snmp trap-group] hierarchy level:

```
snmp {
    trap-group group-name {
        categories [ categories ];
        targets {
            address;
        }
    }
}
```

Example: Setting Trap Notification for Remote Operations

Specify 172.17.12.213 as a target host for all remote operation traps:

```
snmp {
    trap-group remote-traps {
        categories remote-operations;
        targets {
            172.17.12.213;
        }
    }
}
```

For more information on trap groups, see “Configuring SNMP Trap Groups” on page 35.

Using Variable-Length String Indexes

All tabular objects in the remote operations MIBs supported by JUNOS are indexed by two variables of type SnmpAdminString. For more information on SnmpAdminString, see RFC 2571.

JUNOS does not handle SnmpAdminString any differently from the octet string variable type. However, the indexes are defined as variable length. When a variable length string is used as an index, the length of the string must be included as part of the OID.

Example: Set Variable-Length String Indexes

To reference the pingCtlTargetAddress variable of a row in pingCtlTable where pingCtlOwnerIndex is bob and pingCtlTestName is test, use the following OID:

```
pingMIB.pingObjects.pingCtlTable.pingCtlEntry.pingCtlTargetAddress.”bob”.”test”
1.3.6.1.2.1.80.1.2.1.4.3.98.111.98.4.116.101.115.116
```

For more information on the definition of the ping MIB, see RFC 2925.

Enabling Logging

The SNMP error code returned in response to SNMP requests can only provide a generic description of the problem. The error descriptions logged by the remote operations daemon can often provide more detailed information on the problem and help you to solve the problem faster. This logging is not enabled by default. To enable logging, include the flag general statement at the [edit snmp traceoptions] hierarchy level:

```
snmp {
    traceoptions {
        flag general;
    }
}
```

For more information on traceoptions, see "Tracing SNMP Activity" on page 40.

If the remote operations daemon receives an SNMP request that it cannot accommodate, the error is logged in the /var/log/rmopd file. To monitor this log file, issue the monitor start rmopd command in operational mode of the command-line interface (CLI).

Using the Ping MIB

A ping test is used to determine whether packets sent from the local host reach the designated host and are returned. If the designated host can be reached, the ping test provides the approximate round-trip time for the packets. Ping test results are stored in pingResultsTable and pingProbeHistoryTable.

RFC 2925 is the authoritative description of the ping MIB in detail and provides the ASN.1 MIB definition of the ping MIB. This section includes the following topics:

Starting a Ping Test

Before you start a ping test, configure a ping MIB view. This allows SNMP Set requests on pingMIB. To start a ping test, create a row in pingCtlTable and set pingCtlAdminStatus to enabled. The minimum information that must be specified before setting pingCtlAdminStatus to enabled is:

- pingCtlOwnerIndexSnmpAdminString
- pingCtlTestNameSnmpAdminString
- pingCtlTargetAddressInetAddress
- pingCtlTargetAddressTypeInetAddressType
- pingCtlRowStatusRowStatus

For all other values, defaults are chosen unless otherwise specified. pingCtlOwnerIndex and pingCtlTestName are used as the index, so their values are specified as part of the OID. To create a row, set pingCtlRowStatus to createAndWait or createAndGo on a row that does not already exist. A value of active for pingCtlRowStatus indicates that all necessary information has been supplied and the test can begin; pingCtlAdminStatus can be set to enabled. An SNMP Set request that sets pingCtlRowStatus to active will fail if the necessary information in the row is not specified or is inconsistent. For information about how to configure a view, see "Setting SNMP Views" on page 84.

There are two ways to start a ping test:

- Using Multiple Set PDUs on page 88
- Using a Single Set PDU on page 89

Using Multiple Set PDUs

You can use multiple Set request PDUs (multiple PDUs, with one or more varbind each) and set the following variables in this order to start the test:

- pingCtlRowStatus to createAndWait
- All appropriate test variables
- pingCtlRowStatus to active

  The JUNOS software now verifies that all necessary information to run a test has been specified.

- pingCtlAdminStatus to enabled

Using a Single Set PDU

You can use a single Set request PDU (one PDU, with multiple varbinds) to set the following variables to start the test:

- pingCtlRowStatus to createAndGo
- All appropriate test variables
- pingCtlAdminStatus to enabled

Monitoring a Running Ping Test

When pingCtlAdminStatus is successfully set to enabled, the following is done before the acknowledgement of the SNMP Set request is sent back to the client:

- pingResultsEntry is created if it does not already exist.
- pingResultsOperStatus transitions to enabled.

pingResultsTable

While the test is running, pingResultsEntry keeps track of the status of the test. The value of pingResultsOperStatus is enabled while the test is running and disabled when it has stopped.

The value of pingCtlAdminStatus remains enabled until you set it to disabled. Thus, to get the status of the test, you must examine pingResultsOperStatus.

The pingCtlFrequency variable can be used to schedule many tests for one pingCtlEntry. After a test ends normally (you did not stop the test) and pingCtlFrequency number of seconds has elapsed, the test is started again just as if you had set pingCtlAdminStatus to enabled. If you intervene at any time between repeated tests (you set pingCtlAdminStatus to disabled or pingCtlRowStatus to notInService), the repeat feature is disabled until another test is started and ends normally. A value of 0 for pingCtlFrequency indicates this repeat feature is not active.

pingResultsIpTgtAddr and pingResultsIpTgtAddrType are set to the value of the resolved destination address when the value of pingCtlTargetAddressType is dns. When a test starts successfully and pingResultsOperStatus transitions to enabled:

- pingResultsIpTgtAddr is set to null-string.
- pingResultsIpTgtAddrType is set to unknown.

pingResultsIpTgtAddr and pingResultsIpTgtAddrType are not set until pingCtlTargetAddress can be resolved to a numeric address.To retrieve these values, poll pingResultsIpTgtAddrType for any value other than unknown after successfully setting pingCtlAdminStatus to enabled.

At the start of a test, pingResultsSentProbes is initialized to 1 and the first probe is sent. pingResultsSentProbes increases by 1 each time a probe is sent.

As the test runs, every pingCtlTimeOut seconds, the following occurs:

- pingProbeHistoryStatus for the corresponding pingProbeHistoryEntry in pingProbeHistoryTable is set to requestTimedOut.
- A pingProbeFailed trap is generated, if necessary.
- An attempt is made to send the next probe.

NOTE: No more than one outstanding probe exists for each test.

For every probe, you can receive one of the following results:

- The target host acknowledges the probe with a response.
- The probe times out; there is no response from the target host acknowledging the probe.
- The probe could not be sent.

Each probe result is recorded in pingProbeHistoryTable. For more information on pingProbeHistoryTable, see pingProbeHistoryTable on page 91.

When a response is received from the target host acknowledging the current probe:

- pingResultsProbeResponses increases by 1.
- The following variables are updated:
  - pingResultsMinRtt—Minimum round-trip time
  - pingResultsMaxRtt—Maximum round-trip time
  - pingResultsAverageRtt—Average round-trip time
  - pingResultsRttSumOfSquares—Sum of squares of round-trip times
  - pingResultsLastGoodProbe—Timestamp of the last response

NOTE: Only probes that result in a response from the target host contribute to the calculation of the round-trip time (rtt) variables.

When a response to the last probe is received or the last probe has timed out, the test is complete.

pingProbeHistoryTable

An entry in pingProbeHistoryTable (pingProbeHistoryEntry) represents a probe result and is indexed by three variables:

- The first two variables, pingCtlOwnerIndex and pingCtlTestName, are the same ones used for pingCtlTable, which identifies the test.
- The third variable, pingProbeHistoryIndex, is a counter to uniquely identify each probe result.

The maximum number of pingProbeHistoryTable entries created for a given test is limited by pingCtlMaxRows. If pingCtlMaxRows is set to 0, no pingProbeHistoryTable entries will be created for that test.

Each time a probe result is determined, a pingProbeHistoryEntry is created and added to pingProbeHistoryTable. pingProbeHistoryIndex of the new pingProbeHistoryEntry is 1 greater than the last pingProbeHistoryEntry added to pingProbeHistoryTable for that test. pingProbeHistoryIndex is set to 1 if this is the first entry in the table. The same test can be run multiple times, so this index keeps growing.

If pingProbeHistoryIndex of the last pingProbeHistoryEntry added is 0xFFFFFFFF, the next pingProbeHistoryEntry added has pingProbeHistoryIndex set to 1.

The following is recorded for each probe result:

- pingProbeHistoryResponse—Time to live (TTL)
- pingProbeHistoryStatus—What happened and why
- pingProbeHistoryLastRC—Return code (RC) value of ICMP packet
- pingProbeHistoryTime—Timestamp when probe result was determined

When a probe cannot be sent, pingProbeHistoryResponse is set to 0. When a probe times out, pingProbeHistoryResponse is set to the difference between the time when the probe was discovered to be timed out and the time when the probe was sent.

Generating Traps

For any trap to be generated, the appropriate bit of pingCtlTrapGeneration must be set. You must also configure a trap group to receive remote operations. A trap is generated under the following conditions:

- A pingProbeFailed trap is generated every time pingCtlTrapProbeFailureFilter number of consecutive probes fail during the test.
- A pingTestFailed trap is generated when the test completes and at least pingCtlTrapTestFailureFilter number of probes fail.
- A pingTestCompleted trap is generated when the test completes and fewer than pingCtlTrapTestFailureFilter probes fail.

NOTE: A probe is considered a failure when pingProbeHistoryStatus of the probe result is anything besides responseReceived.

For information about how to configure a trap group to receive remote operations, see "Configuring SNMP Trap Groups" on page 35 and "Example: Setting Trap Notification for Remote Operations" on page 86.

Gathering Ping Test Results

You can either poll pingResultsOperStatus to find out when the test is complete or request to get a trap when the test is complete. For more information on pingResultsOperStatus, see pingResultsTable on page 89. For more information on ping MIB traps, see "Generating Traps" on page 92.

The statistics calculated and then stored in pingResultsTable include:

- pingResultsMinRtt—Minimum round-trip time
- pingResultsMaxRtt—Maximum round-trip time
- pingResultsAverageRtt—Average round-trip time
- pingResultsProbeResponses—Number of responses received
- pingResultsSentProbes—Number of attempts to send probes
- pingResultsRttSumOfSquares—Sum of squares of round-trip times
- pingResultsLastGoodProbe—Timestamp of the last response

You can also consult pingProbeHistoryTable for more detailed information on each probe. The index used for pingProbeHistoryTable starts at 1, goes to 0xFFFFFFFF, and wraps to 1 again.

For example, if pingCtlProbeCount is 15 and pingCtlMaxRows is 5, then upon completion of the first run of this test, pingProbeHistoryTable will contain probes like those in Table 6.

Table 6: Results in pingProbeHistoryTable: After the First Ping Test

pingProbeHistoryIndex	Probe Result
11	Result of 11th probe from run 1
12	Result of 12th probe from run 1
13	Result of 13th probe from run 1
14	Result of 14th probe from run 1
15	Result of 15th probe from run 1

Upon completion of the first probe of the second run of this test, pingProbeHistoryTable will contain probes like those in Table 7.

Table 7: Results in pingProbeHistoryTable: After the First Probe of the Second Test

pingProbeHistoryIndex	Probe Result
12	Result of 12th probe from run 1
13	Result of 13th probe from run 1
14	Result of 14th probe from run 1
15	Result of 15th probe from run 1
16	Result of 1st probe from run 2

Upon completion of the second run of this test, pingProbeHistoryTable will contain probes like those in Table 8.

Table 8: Results in pingProbeHistoryTable: After the Second Ping Test

pingProbeHistoryIndex	Probe Result
26	Result of 11th probe from run 2
27	Result of 12th probe from run 2
28	Result of 13th probe from run 2
29	Result of 14th probe from run 2
30	Result of 15th probe from run 2

History entries can be deleted from the MIB in two ways:

- More history entries for a given test are added and the number of history entries exceeds pingCtlMaxRows. The oldest history entries are deleted to make room for the new ones.
- You delete the entire test by setting pingCtlRowStatus to destroy.

Stopping a Ping Test

To stop an active test, set pingCtlAdminStatus to disabled. To stop the test and remove its pingCtlEntry, pingResultsEntry, and any pingHistoryEntry objects from the MIB, set pingCtlRowStatus to destroy.

Interpreting Ping Variables

This section clarifies the ranges for the following variables that are not explicitly specified in the ping MIB:

- pingCtlDataSize—The value of this variable represents the total size of the payload (in bytes) of an outgoing probe packet. This payload includes the timestamp (8 bytes) that is used to time the probe. This is consistent with the definition of pingCtlDataSize (maximum value of 65,507) and the standard ping application.

  If the value of pingCtlDataSize is between 0 and 8 inclusive, it is ignored and the payload is 8 bytes (the timestamp). The ping MIB assumes all probes are timed, so the payload must always include the timestamp.

  For example, if you wish to add an additional 4 bytes of payload to the packet, you must set pingCtlDataSize to 12.

- pingCtlDataFill—The first 8 bytes of the data segment of the packet is for the timestamp. After that, the pingCtlDataFill pattern is used in repetition. The default pattern (when pingCtlDataFill is not specified) is (00, 01, 02, 03 ... FF, 00, 01, 02, 03 ... FF, ...).

- pingCtlMaxRows—The maximum value is 255.

- pingMaxConcurrentRequests—The maximum value is 500.

- pingCtlTrapProbeFailureFilter and pingCtlTrapTestFailureFilter—A value of 0 for pingCtlTrapProbeFailureFilter or pingCtlTrapTestFailureFilter is not well defined by the ping MIB. If pingCtlTrapProbeFailureFilter is 0, pingProbeFailed traps will not be generated for the test under any circumstances. If pingCtlTrapTestFailureFilter is 0, pingTestFailed traps will not be generated for the test under any circumstances.

Using the Traceroute MIB

A traceroute test approximates the path packets take from the local host to the remote host.

RFC 2925 is the authoritative description of the traceroute MIB in detail and provides the ASN.1 MIB definition of the traceroute MIB. This section provides the following information:

Starting a Traceroute Test

Before you start a traceroute test, configure a traceroute MIB view. This allows SNMP **Set** requests on **tracerouteMIB**. To start a test, create a row in **traceRouteCtlTable** and set **traceRouteCtlAdminStatus** to **enabled**. You must specify at least the following before setting **traceRouteCtlAdminStatus** to **enabled**:

- **traceRouteCtlOwnerIndexSnmpAdminString**
- **traceRouteCtlTestNameSnmpAdminString**
- **traceRouteCtlTargetAddressInetAddress**
- **traceRouteCtlRowStatusRowStatus**

For all other values, defaults are chosen unless otherwise specified. **traceRouteCtlOwnerIndex** and **traceRouteCtlTestName** are used as the index, so their values are specified as part of the OID. To create a row, set **traceRouteCtlRowStatus** to **createAndWait** or **createAndGo** on a row that does not already exist. A value of **active** for **traceRouteCtlRowStatus** indicates that all necessary information has been specified and the test can begin; **traceRouteCtlAdminStatus** can be set to **enabled**. An SNMP **Set** request that sets **traceRouteCtlRowStatus** to **active** will fail if the necessary information in the row is not specified or is inconsistent. For information about how to configure a view, see “Setting SNMP Views” on page 84.

There are two ways to start a traceroute test:

Using Multiple Set PDUs

You can use multiple Set request PDUs (multiple PDUs, with one or more varbind each) and set the following variables in this order to start the test:

- traceRouteCtlRowStatus to createAndWait
- All appropriate test variables
- traceRouteCtlRowStatus to active

  The JUNOS software now verifies that all necessary information to run a test has been specified.
- traceRouteCtlAdminStatus to enabled

Using a Single Set PDU

You can use a single Set request PDU (one PDU, with multiple varbinds) to set the following variables to start the test:

- traceRouteCtlRowStatus to createAndGo
- All appropriate test variables
- traceRouteCtlAdminStatus to enabled

Monitoring a Running Traceroute Test

When traceRouteCtlAdminStatus is successfully set to enabled, the following is done before the acknowledgement of the SNMP Set request is sent back to the client:

- traceRouteResultsEntry is created if it does not already exist.
- traceRouteResultsOperStatus transitions to enabled.

traceRouteResultsTable

While the test is running, this traceRouteResultsTable keeps track of the status of the test. The value of traceRouteResultsOperStatus is enabled while the test is running and disabled when it has stopped.

The value of traceRouteCtlAdminStatus remains enabled until you set it to disabled. Thus, to get the status of the test, you must examine traceRouteResultsOperStatus.

The traceRouteCtlFrequency variable can be used to schedule many tests for one traceRouteCtlEntry. After a test ends normally (you did not stop the test) and traceRouteCtlFrequency number of seconds has elapsed, the test is started again just as if you had set traceRouteCtlAdminStatus to enabled. If you intervene at any time between repeated tests (you set traceRouteCtlAdminStatus to disabled or traceRouteCtlRowStatus to notInService), the repeat feature will be disabled until another test is started and ends normally. A value of 0 for traceRouteCtlFrequency indicates this repeat feature is not active.

traceRouteResultsIpTgtAddr and traceRouteResultsIpTgtAddrType are set to the value of the resolved destination address when the value of traceRouteCtlTargetAddressType is dns. When a test starts successfully and traceRouteResultsOperStatus transitions to enabled:

- traceRouteResultsIpTgtAddr is set to null-string.
- traceRouteResultsIpTgtAddrType is set to unknown.

traceRouteResultsIpTgtAddr and traceRouteResultsIpTgtAddrType are not set until traceRouteCtlTargetAddress can be resolved to a numeric address. To retrieve these values, poll traceRouteResultsIpTgtAddrType for any value other than unknown after successfully setting traceRouteCtlAdminStatus to enabled.

At the start of a test, traceRouteResultsCurHopCount is initialized to traceRouteCtlInitialTtl, and traceRouteResultsCurProbeCount is initialized to 1. Each time a probe result is determined, traceRouteResultsCurProbeCount increases by 1. While the test is running, the value of traceRouteResultsCurProbeCount reflects the current outstanding probe for which results have not yet been determined.

traceRouteCtlProbesPerHop number of probes are sent for each TTL value. When the result of the last probe for the current hop is determined, provided that the current hop is not the destination hop, traceRouteResultsCurHopCount increases by 1, and traceRouteResultsCurProbeCount resets to 1.

At the start of a test, if this is the first time this test has been run for this traceRouteCtlEntry, traceRouteResultsTestAttempts and traceRouteResultsTestSuccesses are initialized to 0.

At the end of each test execution, traceRouteResultsOperStatus transitions to disabled, and traceRouteResultsTestAttempts increases by 1. If the test was successful in determining the full path to the target, traceRouteResultsTestSuccesses increases by 1, and traceRouteResultsLastGoodPath is set to the current time.

traceRouteProbeResultsTable

Each entry in traceRouteProbeHistoryTable is indexed by five variables:

- The first two variables, traceRouteCtlOwnerIndex and traceRouteCtlTestName, are the same ones used for traceRouteCtlTable and to identify the test.
- The third variable, traceRouteProbeHistoryIndex, is a counter, starting from 1 and wrapping at FFFFFFFF. The maximum number of entries is limited by traceRouteCtlMaxRows.
- The fourth variable, traceRouteProbeHistoryHopIndex, indicates which hop this probe is for (the actual TTL value). Thus, the first traceRouteCtlProbesPerHop number of entries created when a test starts have a value of traceRouteCtlInitialTtl for traceRouteProbeHistoryHopIndex.
- The fifth variable, traceRouteProbeHistoryProbeIndex, is the probe for the current hop. It ranges from 1 to traceRouteCtlProbesPerHop.

While a test is running, as soon as a probe result is determined, the next probe is sent. A maximum of traceRouteCtlTimeOut seconds elapses before a probe is marked with status requestTimedOut and the next probe is sent. There is never more than one outstanding probe per traceroute test. Any probe result coming back after a probe times out is ignored.

Each probe can:

- Result in a response from a host acknowledging the probe
- Time out with no response from a host acknowledging the probe
- Fail to be sent

Each probe status is recorded in traceRouteProbeHistoryTable with traceRouteProbeHistoryStatus set accordingly.

Probes that result in a response from a host record the following data:

- traceRouteProbeHistoryResponse—Round-trip time (rtt)
- traceRouteProbeHistoryHAddrType—The type of HAddr (next argument)
- traceRouteProbeHistoryHAddr—The address of the hop

All probes, regardless of whether a response for the probe is received, have the following recorded:

- traceRouteProbeHistoryStatus—What happened and why
- traceRouteProbeHistoryLastRC—Return code (RC) value of the ICMP packet
- traceRouteProbeHistoryTime—Timestamp when the probe result was determined

When a probe cannot be sent, traceRouteProbeHistoryResponse is set to 0. When a probe times out, traceRouteProbeHistoryResponse is set to the difference between the time when the probe was discovered to be timed out and the time when the probe was sent.

traceRouteHopsTable

Entries in traceRouteHopsTable are indexed by three variables:

- The first two, traceRouteCtlOwnerIndex and traceRouteCtlTestName, are the same ones used for traceRouteCtlTable and identify the test.
- The third variable, traceRouteHopsHopIndex, indicates the current hop, which starts at 1 (not traceRouteCtlInitialTtl).

When a test starts, all entries in traceRouteHopsTable with the given traceRouteCtlOwnerIndex and traceRouteCtlTestName are deleted. Entries in this table are only created if traceRouteCtlCreateHopsEntries is set to true.

A new traceRouteHopsEntry is created each time the first probe result for a given TTL is determined. The new entry is created whether or not the first probe reaches a host. The value of traceRouteHopsHopIndex is increased by 1 for this new entry.

NOTE: Any traceRouteHopsEntry can lack a value for traceRouteHopsIpTgtAddress if there are no responses to the probes with the given TTL.

Each time a probe reaches a host, the IP address of that host is available in the probe result. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is not set, then the value of traceRouteHopsIpTgtAddress is set to this IP address. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is the same as the IP address, then the value does not change. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is different from this IP address, indicating a path change, a new traceRouteHopsEntry is created with:

- traceRouteHopsHopIndex variable increased by 1
- traceRouteHopsIpTgtAddress set to the IP address

NOTE: A new entry for a test is added to traceRouteHopsTable each time a new TTL value is used or the path changes. Thus, the number of entries for a test may exceed the number of different TTL values used.

When a probe result is determined, the value traceRouteHopsSentProbes of the current traceRouteHopsEntry increases by 1. When a probe result is determined, and the probe reaches a host:

- The value traceRouteHopsProbeResponses of the current traceRouteHopsEntry is increased by 1.
- The following variables are updated:
  - traceRouteResultsMinRtt—Minimum round-trip time
  - traceRouteResultsMaxRtt—Maximum round-trip time
  - traceRouteResultsAverageRtt—Average round-trip time
  - traceRouteResultsRttSumOfSquares—Sum of squares of round-trip times
  - traceRouteResultsLastGoodProbe—Timestamp of the last response

NOTE: Only probes that reach a host affect the round-trip time values.

Generating Traps

For any trap to be generated, the appropriate bit of traceRouteCtlTrapGeneration must be set. You must also configure a trap group to receive remote operations. Traps are generated under the following conditions:

- traceRouteHopsIpTgtAddress of the current probe is different from the last probe with the same TTL value (traceRoutePathChange).
- A path to the target could not be determined (traceRouteTestFailed).

A path to the target was determined (traceRouteTestCompleted).

For information about how to configure a trap group to receive remote operations, see “Configuring SNMP Trap Groups” on page 35 and “Example: Setting Trap Notification for Remote Operations” on page 86.

Monitoring Traceroute Test Completion

When a test is complete, traceRouteResultsOperStatus transitions from enabled to disabled. This transition occurs in the following situations:

- The test ends successfully. A probe result indicates that the destination has been reached. In this case, the current hop is the last hop. The rest of the probes for this hop are sent. When the last probe result for the current hop is determined, the test ends.

- traceRouteCtlMaxTtl threshold is exceeded. The destination is never reached. The test ends after the number of probes with TTL value equal to traceRouteCtlMaxttl have been sent.

- traceRouteCtlMaxFailures threshold is exceeded. The number of consecutive probes that end with status requestTimedOut exceeds traceRouteCtlMaxFailures.

- You end the test. You set traceRouteCtlAdminStatus to disabled or delete the row by setting traceRouteCtlRowStatus to destroy.

- You misconfigured the traceroute test. A value or variable you specified in traceRouteCtlTable is incorrect and will not allow a single probe to be sent. Because of the nature of the data, this error could not be determined until the test was started; that is, until after traceRouteResultsOperStatus transitioned to enabled. When this occurs, one entry is added to traceRouteProbeHistoryTable with traceRouteProbeHistoryStatus set to the appropriate error code.

If traceRouteCtlTrapGeneration is set properly, either the traceRouteTestFailed or traceRouteTestCompleted trap is generated.

Gathering Traceroute Test Results

You can either poll traceRouteResultsOperStatus to find out when the test is complete or request to get a trap when the test is complete. For more information on traceResultsOperStatus, see traceRouteResultsTable on page 97. For more information on traceroute MIB traps, see "Generating Traps" on page 100.

Statistics are calculated on a per-hop basis and then stored in traceRouteHopsTable. They include the following for each hop:

- traceRouteHopsIpTgtAddressType—Address type of host at this hop
- traceRouteHopsIpTgtAddress—Address of host at this hop
- traceRouteHopsMinRtt—Minimum round-trip time
- traceRouteHopsMaxRtt—Maximum round-trip time
- traceRouteHopsAverageRtt—Average round-trip time
- traceRouteHopsRttSumOfSquares—Sum of squares of round-trip times
- traceRouteHopsSentProbes—Number of attempts to send probes
- traceRouteHopsProbeResponses—Number of responses received
- traceRouteHopsLastGoodProbe—Timestamp of last response

You can also consult traceRouteProbeHistoryTable for more detailed information on each probe. The index used for traceRouteProbeHistoryTable starts at 1, goes to 0xFFFFFFFF, and wraps to 1 again.

For example, assume the following:

- traceRouteCtlMaxRows is 10.
- traceRouteCtlProbesPerHop is 5.
- There are 8 hops to the target (the target being number 8).
- Each probe sent results in a response from a host (the number of probes sent is not limited by traceRouteCtlMaxFailures).

In this test, 40 probes are sent. At the end of the test, traceRouteProbeHistoryTable would have a history of probes like those in Table 9.

Table 9: traceRouteProbeHistoryTable

HistoryIndex	HistoryHopIndex	HistoryProbeIndex
31	7	1
32	7	2
33	7	3
34	7	4
35	7	5
36	8	1
37	8	2
38	8	3
39	8	4
40	8	5

Stopping a Traceroute Test

To stop an active test, set traceRouteCtlAdminStatus to disabled. To stop a test and remove its traceRouteCtlEntry, traceRouteResultsEntry, traceRouteProbeHistoryEntry, and traceRouteProbeHistoryEntry objects from the MIB, set traceRouteCtlRowStatus to destroy.

Traceroute Variables

This section clarifies the ranges for the following variables that are not explicitly specified in the traceroute MIB:

- traceRouteCtlMaxRows—The maximum value for traceRouteCtlMaxRows is 2550. This represents the maximum TTL (255) multiplied by the maximum for traceRouteCtlProbesPerHop (10). Therefore, the traceRouteProbeHistoryTable accommodates one complete test at the maximum values for one traceRouteCtlEntry. Usually, the maximum values are not used and the traceRouteProbeHistoryTable is able to accommodate the complete history for many tests for the same traceRouteCtlEntry.

- traceRouteMaxConcurrentRequests—The maximum value is 50. If a test is running, it has one outstanding probe. traceRouteMaxConcurrentRequests represents the maximum number of traceroute tests that have traceRouteResultsOperStatus with a value of enabled. Any attempt to start a test with traceRouteMaxConcurrentRequests tests running will result in the creation of one probe with traceRouteProbeHistoryStatus set to maxConcurrentLimitReached and that test will end immediately.

- traceRouteCtlTable—The maximum number of entries allowed in this table is 100. Any attempt to create a 101st entry will result in a BAD_VALUE message for SNMPv1 and a RESOURCE_UNAVAILABLE message for SNMPv2.

Chapter 9

Juniper Networks Enterprise-Specific MIBs

The JUNOS software supports the following enterprise-specific Management Information Bases (MIBs):

- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chassis-alarm.txt.

- ATM CoS MIB—Provides support for monitoring Asynchronous Transfer Mode, version 2 (ATM2) virtual circuit (VC) class of service (CoS) configurations. It also provides CoS queue statistics for all VCs that have CoS configured. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-atm-cos.txt.

- ATM MIB—Provides support for ATM interfaces and virtual connections. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74net-mgmt/html/mib-jnx-atm.txt.

- BGP4 V2 MIB—Contains objects used to monitor Border Gateway Protocol (BGP) peer-received prefix counters. It is based upon similar objects in the MIB documented in Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version*. For more information about the BGP4 V2 MIB, see "Interpreting the Enterprise-Specific BGP4 V2 MIB" on page 329. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-bgpmib2.txt.

- Chassis MIB—Provides support for environmental monitoring (power supply state, board voltages, fans, temperatures, and air flow) and inventory support for the chassis, System Control Board (SCB), System and Switching Board (SSB), Switching and Forwarding Model (SFM), Flexible PIC Concentrators (FPCs), and Physical Interface Cards (PICs). For more information about the chassis MIB, see "Interpreting the Enterprise-Specific Chassis MIBs" on page 243. For a downloadable version, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chassis.txt.

- Chassis Definitions for Router Model MIB—Contains the object identifiers (OIDs) that are used by the chassis MIB to identify platform and chassis components. The chassis MIB provides information that changes often, while the chassis definitions for router model MIB provide information that changes less often. For more information about the chassis definitions for the router model MIB, see "Chassis Definitions for the Router Model MIB" on page 326. For a downloadable version of the chassis definitions for router mode MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chas-defines.txt.

- Class of Service MIB—Provides support for monitoring interface output queue statistics per interface and per forwarding class. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-cos.txt.

- Configuration Management MIB—Provides notification for configuration changes as 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. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-cfgmgmt.txt.

- Destination Class Usage MIB—Provides support for monitoring packet counts based on the ingress and egress points for traffic transiting your networks. Ingress points are identified by input interface. Egress points are identified by destination prefixes grouped into one or more sets, known as destination classes. One counter is managed per interface per destination class, up to a maximum of 16 counters per interface. For information about the destination class usage MIB, see "Interpreting the Enterprise-Specific Destination Class Usage MIB" on page 327. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-dcu.txt.

- Ethernet MAC MIB—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. For more information about the Ethernet MAC MIB, see the "Interpreting the Enterprise-Specific Ethernet MAC MIB" on page 365. To view this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-mac.txt.

- Experimental MIB—Contains object identifiers for experimental MIBs. To view this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-exp.txt.

- Firewall MIB—Provides support for monitoring firewall filter counters. Routers must have the Internet Processor II ASIC to perform firewall monitoring. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-firewall.txt.

- Flow Collection Services MIB—Provides statistics on files, records, memory, FTP, and error states of a monitoring services interface. It also provides SNMP traps for unavailable destinations, unsuccessful file transfers, flow overloading, and memory overloading. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-coll.txt.

- Host Resources MIB—Extends the hrStorageTable object, providing a measure of the usage of each file system on the router in 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. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-hostresources.txt.

- Interface MIB—Extends the standard ifTable (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information. For more information about this MIB, see "Interpreting the Enterprise-Specific Interface MIB" on page 367. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-if-extensions.txt.

- IPv4 MIB—Provides additional Internet Protocol version 4 (IPv4) address information, supporting the assignment of identical IPv4 addresses to separate interfaces. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-ipv4.txt.

- IPv6 and ICMPv6 MIB—Provides IPv6 and Internet Control Message Protocol version 6 (ICMPv6) statistics. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-ipv6.txt.

- LDP MIB—Provides Label Distribution Protocol (LDP) statistics and defines LDP label-switched path (LSP) notifications. LDP traps support only IPv4 standards. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-ldp.txt.

- MPLS MIB—Provides Multiprotocol Label Switching (MPLS) information and defines MPLS notifications. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-mpls.txt.

- Passive Monitoring MIB—Performs traffic flow monitoring and lawful interception of packets transiting between two routers. For more information about the passive monitoring MIB, see the *JUNOS Feature Guide* and "Interpreting the Enterprise-Specific Passive Monitoring MIB" on page 351. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-pmon.txt.

- Ping MIB— 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. For more information about the ping MIB, see "Interpreting the Enterprise-Specific Ping MIB" on page 331. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-ping.txt.

- Resource Reservation Protocol (RSVP) traffic engineering (TE) MIB—Provides information about RSVP-TE sessions that correspond to MPLS LSPs on transit routing platforms in the service provider core network. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-rsvp.txt.

- Reverse-Path-Forwarding MIB—Monitors statistics for traffic that is rejected because of reverse-path-forwarding (RPF) processing. For more information about the reverse-path-forwarding MIB, see "Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB" on page 347. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-rpf.txt.

- RMON Events and Alarms MIB—Supports the JUNOS extensions to the standard remote monitoring (RMON) events and alarms MIB (RFC 2819). The extension augments alarmTable with additional information about each alarm. Two new traps are also defined to indicate when problems are encountered with an alarm. For more information about the RMON events and alarms MIB, see "Interpreting the Enterprise-Specific RMON Events and Alarms MIB" on page 343. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-rmon.txt.

- The Services PIC MIB—Provides statistics for Adaptive Services (AS) PICs and defines notifications for AS PICs. For more information about the Services PIC MIB, see "Interpreting the Enterprise-Specific Services PIC MIB" on page 389. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sp.txt.

- SONET/SDH Interface Management MIB—Monitors the current alarm for each SONET/SDH interface. For more information about the SONET interface management MIB, see "Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB" on page 353. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sonet.txt.

- SONET Automatic Protection Switching MIB—Monitors any SONET interface that participates in Automatic Protection Switching (APS). For more information about the SONET APS MIB, see "Interpreting the Enterprise-Specific SONET APS MIB" on page 355. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sonetaps.txt.

- Source Class Usage MIB—Counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. The source class usage (SCU) MIB makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. For more information about the source class usage MIB, see "Interpreting the Enterprise-Specific Source Class Usage MIB" on page 349. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-scu.txt.

- Structure of Management Information MIB—Explains how the Juniper Networks enterprise-specific MIBs are structured. For more information on the structure of management information (SMI) MIB, see “Interpreting the Structure of Management Information MIB” on page 239. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-smi.txt.

- Traceroute MIB—Supports the JUNOS extensions of traceroutes and remote operations. Items in this MIB are created when entries are created in the traceRouteCtlTable of the traceroute MIB. Each item is indexed exactly the same way as it is in the traceroute MIB. For more information about the traceroute MIB, see “Interpreting the Enterprise-Specific Traceroute MIB” on page 341. For a downloadable version, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-traceroute.txt.

- VPN MIB—Provides monitoring for Layer 3 VPNs, Layer 2 VPNs, and virtual private LAN service (VPLS) (read access only). For more information about the VPN MIB, see “Interpreting the Enterprise-Specific VPN MIB” on page 373. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-vpn.txt.

Chapter 10

Juniper Networks Enterprise-Specific SNMP Traps

This chapter summarizes the enterprise-specific SNMP traps supported by the JUNOS software. For scalability reasons, the Multiprotocol Label Switching (MPLS) traps are generated by the ingress router only. For information on disabling the generation of MPLS traps, see the *JUNOS MPLS Applications Configuration Guide*.

NOTE: All enterprise-specific SNMP traps supported by the JUNOS software can be sent in version 1 and 2 formats.

The JUNOS software supports the following enterprise-specific traps:

Juniper Networks Enterprise-Specific SNMP Version 1 Traps

The JUNOS software supports enterprise-specific SNMP version 1 traps shown in Table 10. 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. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system log messages, see the *JUNOS System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS System Basics Configuration Guide*. To view the Juniper Networks enterprise-specific SNMP version 1 traps, see "Juniper Networks Enterprise-Specific MIBs" on page 105 and select the corresponding Juniper Networks enterprise-specific MIB. For more information about chassis traps, see "Chassis Traps" on page 322.

Table 10 lists the Juniper Networks enterprise-specific supported SNMP version 1 traps.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
chassis (alarm conditions)	jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1	warning	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2	critical	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3	alert	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1	6	4	critical	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9	warning	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11	notice	CHASSISD_ SNMP_ TRAP
chassis (alarm conditions)	jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12	warning	CHASSISD_ SNMP_ TRAP
chassis (cleared alarm conditions)	jnxPowerSupplyOk	1.3.6.1.4.1.2636.4.2	6	1	critical	CHASSISD_ SNMP_ TRAP
chassis (cleared alarm conditions)	jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2	critical	CHASSISD_ SNMP_ TRAP

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
chassis (cleared alarm conditions)	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3	alert	CHASSISD_ SNMP_ TRAP
configuration	jnxCmCfgChange	1.3.6.1.4.1.2636.4.5	6	1	—	—
configuration	jnxCmRescueChange	1.3.6.1.4.1.2636.4.5	6	2	—	—
link	jnxCollUnavailableDest	1.3.6.1.4.1.2636.4.8	6	1	—	—
link	jnxCollUnavailableDestCleared	1.3.6.1.4.1.2636.4.8	6	2	—	—
link	jnxCollUnsuccessfulTransfer	1.3.6.1.4.1.2636.4.8	6	3	—	—
link	jnxCollFlowOverload	1.3.6.1.4.1.2636.4.8	6	4	—	—
link	jnxCollFlowOverloadCleared	1.3.6.1.4.1.2636.4.8	6	5	—	—
link	jnxCollMemoryUnavailable	1.3.6.1.4.1.2636.4.8	6	6	—	—
link	jnxCollMemoryAvailable	1.3.6.1.4.1.2636.4.8	6	7	—	—
link	jnxPMonOverloadSet	1.3.6.1.4.1.2636.4.7.0.1	6	1	—	—
link	jnxPMonOverloadCleared	1.3.6.1.4.1.2636.4.7.0.2	6	2	—	—
link	jnxapsEventSwitchover	1.3.6.1.4.1.2636.3.24.2	6	1	—	—
link	jnxapsEventModeMismatch	1.3.6.1.4.1.2636.3.24.2	6	2	—	—
link	apsEventChannelMismatch	1.3.6.1.4.1.2636.3.24.2	6	3	—	—
link	apsEventPSBF	1.3.6.1.4.1.2636.3.24.2	6	4	—	—
link	apsEventFEPLF	1.3.6.1.4.1.2636.3.24.2	6	5	—	—
remote operations	jnxPingRttThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	1		
remote operations	jnxPingRttStdDevThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	2		
remote operations	jnxPingRttJitterThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	3		
remote operations	jnxPingEgressThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	4		
remote operations	jnxPingEgressStdDevThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	5		
remote operations	jnxPingEgressJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	6		
remote operations	jnxPingIngressThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	7		
remote operations	jnxPingIngressStddevThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	8		
remote operations	jnxPingIngressJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	9		
routing	jnxBgpM2Established	1.3.6.1.4.1.2636.5.1.1.1.0.1	6	1	—	—
routing	jnxBgpM2BackwardTransition	1.3.6.1.4.1.2636.5.1.1.1.0.2	6	2	—	—
routing	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4	6	1	—	—

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
routing	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4	6	2	—	—
routing	jnxLdpSesUp	1.3.6.1.4.1.2636.4.4	6	3	—	—
routing	jnxLdpSesDown	1.3.6.1.4.1.2636.4.4	6	4	—	—
routing	mplsLspUp	1.3.6.1.4.1.2636.3.2.4	6	1	—	—
routing	mplsLspDown	1.3.6.1.4.1.2636.3.2.4	6	2	—	—
routing	mplsLspChange	1.3.6.1.4.1.2636.3.2.4	6	3	—	—
routing	mplsLspPathDown	1.3.6.1.4.1.2636.3.2.4	6	4	—	—
routing	jnxVpnIfUp	1.3.6.1.4.1.2636.3.26	6	1	—	—
routing	jnxVpnIfDown	1.3.6.1.4.1.2636.3.26	6	2	—	—
routing	jnxVpnPwUp	1.3.6.1.4.1.2636.3.26	6	3	—	—
routing	jnxVpnPwDown	1.3.6.1.4.1.2636.3.26	6	4	—	—
rmon-alarm	jnxRmonAlarmGetFailure	1.3.6.1.4.1.2636.4.3	6	1	—	—
rmon-alarm	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3	6	2	—	—
sonet-alarm	jnxSonetAlarmSet	1.3.6.1.4.1.2636.4.6	6	1	—	—
sonet-alarm	jnxSonetAlarmCleared	1.3.6.1.4.1.2636.4.6	6	2	—	—

Juniper Networks Enterprise-Specific SNMP Version 2 Traps

The JUNOS software supports the enterprise-specific SNMP version 2 traps shown in Table 11. 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. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system messages, see the *JUNOS System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS System Basics Configuration Guide*. To view the Juniper Networks enterprise-specific SNMP version 2 traps, see the "Juniper Networks Enterprise-Specific MIBs" on page 105 and select the corresponding Juniper Networks enterprise-specific MIB. For more information about chassis traps, see "Chassis Traps" on page 322.

Table 11: Enterprise-Specific Supported SNMP Version 2 Traps

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
chassis (alarm conditions)	jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	alert	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	critical	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	critical	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1.4	critical	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	notice	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	notice	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	notice	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	notice	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruFailed	1.3.6.1.4.1.2636.4.1.9	warning	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruOffline	1.3.6.1.4.1.2636.4.1.10	notice	CHASSISD_SNMP_TRAP
chassis (alarm conditions)	jnxFruOnline	1.3.6.1.4.1.2636.4.1.11	notice	CHASSISD_SNMP_TRAP

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
chassis (alarm conditions)	jnxFruCheck	1.3.6.1.4.1.2636.4.1.12	notice	CHASSISD_ SNMP_ TRAP
chassis (cleared alarm conditions)	jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	critical	CHASSISD_ SNMP_ TRAP
chassis (cleared alarm conditions)	jnxFanOK	1.3.6.1.4.1.2636.4.2.2	critical	CHASSISD_ SNMP_ TRAP
chassis (cleared alarm conditions)	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	alert	CHASSISD_ SNMP_ TRAP
configuration	jnxCmCfgChange	1.3.6.1.4.1.2636.4.5.0.1	—	—
configuration	jnxCmRescueChange	1.3.6.1.4.1.2636.4.5.0.2	—	—
link	jnxCollUnavailableDest	1.3.6.1.4.1.2636.4.8.0.1	—	—
link	jnxCollUnavailableDestCleared	1.3.6.1.4.1.2636.4.8.0.2	—	—
link	jnxCollUnsuccessfulTransfer	1.3.6.1.4.1.2636.4.8.0.3	—	—
link	jnxCollFlowOverload	1.3.6.1.4.1.2636.4.8.0.4	—	—
link	jnxCollFlowOverloadCleared	1.3.6.1.4.1.2636.4.8.0.5	—	—
link	jnxCollMemoryUnavailable	1.3.6.1.4.1.2636.4.8.0.6	—	—
link	jnxCollMemoryAvailable	1.3.6.1.4.1.2636.4.8.0.7	—	—
link	jnxPMonOverloadSet	1.3.6.1.4.1.2636.4.7.0.1	—	—
link	jnxPMonOverloadCleared	1.3.6.1.4.1.2636.4.7.0.2	—	—
link	jnxapsEventSwitchover	1.3.6.1.4.1.2636.3.24.2.0.1	—	—
link	jnxapsEventModeMismatch	1.3.6.1.4.1.2636.3.24.2.0.2	—	—
link	apsEventChannelMismatch	1.3.6.1.4.1.2636.3.24.2.0.3	—	—
link	apsEventPSBF	1.3.6.1.4.1.2636.3.24.2.0.4	—	—
link	apsEventFEPLF	1.3.6.1.4.1.2636.3.24.2.0.5	—	—
remote operations	jnxPingRttThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.1		
remote operations	jnxPingRttStdDevThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.2		
remote operations	jnxPingRttJitterThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.3		
remote operations	jnxPingEgressThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.4		
remote operations	jnxPingEgressStdDevThresholdExceed	1.3.6.1.4.1.2636.4.9.0.5		
remote operations	jnxPingEgressJitterThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.6		
remote operations	jnxPingIngressThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.7		
remote operations	jnxPingIngressStddevThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.8		
remote operations	jnxPingIngressJitterThresholdExceedd	1.3.6.1.4.1.2636.4.9.0.9		
routing	jnxBgpM2Established	1.3.6.1.4.1.2636.5.1.1.1.0.1	—	—
routing	jnxBgpM2BackwardTransition	1.3.6.1.4.1.2636.5.1.1.1.0.2	—	—
routing	jnxLdpLspUp	1.3.6.1.4.1.2626.4.4.0.1	—	—
routing	jnxLdpLspDown	1.3.6.1.4.1.2626.4.4.0.2	—	—

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
routing	jnxLdpSesUp	1.3.6.1.4.1.2626.4.4.0.3	—	—
routing	jnxLdpSesDown	1.3.6.1.4.1.2626.4.4.0.4	—	—
routing	mplsLspUp	1.3.6.1.4.1.2636.3.2.4.1	—	—
routing	mplsLspDown	1.3.6.1.4.1.2636.3.2.4.2	—	—
routing	mplsLspChange	1.3.6.1.4.1.2636.3.2.4.3	—	—
routing	mplsLspPathDown	1.3.6.1.4.1.2636.3.2.4.4	—	—
routing	jnxVpnIfUp	1.3.6.1.4.1.2636.3.26.0.1	—	—
routing	jnxVpnIfDown	1.3.6.1.4.1.2636.3.26.0.2	—	—
routing	jnxVpnPwUp	1.3.6.1.4.1.2636.3.26.0.3	—	—
routing	jnxVpnPwDown	1.3.6.1.4.1.2636.3.26.0.4	—	—
rmon-alarm	jnxRmonAlarmGetFailure	1.3.6.1.4.1.2636.4.3.0.1	—	—
rmon-alarm	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3.0.2	—	—
sonet-alarm	jnxSonetAlarmSet	1.3.6.1.4.1.2636.4.6.0.1	—	—
sonet-alarm	jnxSonetAlarmCleared	1.3.6.1.4.1.2636.4.6.0.2	—	—

Chapter 11

Standard SNMP Traps

This chapter summarizes the standard SNMP traps supported by the JUNOS software. For scalability reasons, the Multiprotocol Label Switching (MPLS) traps are generated by the ingress router only. For information on disabling the generation of MPLS traps, see the *JUNOS MPLS Applications Configuration Guide*.

The JUNOS software supports the following standard SNMP traps:

Standard SNMP Version 1 Traps

Table 12 on page 120 provides an overview of the standard traps for SNMPv1. The traps are organized first by trap category and then by trap name, and include their enterprise ID, generic trap number, and specific trap number. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information on system log messages, see the *JUNOS System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS System Basics Configuration Guide*.

Table 12: Standard Supported SNMP Version 1 Traps

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag
authentication	authenticationFailure	1.3.6.1.4.1.2636	4	0	notice	SNMPD_TRAP_ GEN_ FAILURE
link	linkDown	1.3.6.1.4.1.2636	2	0	info	SNMP_TRAP_ LINK_DOWN
link	linkUp	1.3.6.1.4.1.2636	3	0	warning	SNMP_TRAP_ LINK_UP
remote-operations	pingProbeFailed	1.3.6.1.2.1.80.0	6	1	info	SNMP_TRAP_ PING_ PROBE_FAILED
remote-operations	pingTestFailed	1.3.6.1.2.1.80.0	6	2	info	SNMP_TRAP_PING_ TEST_FAILED
remote-operations	pingTestCompleted	1.3.6.1.2.1.80.0	6	3	info	SNMP_TRAP_PING_ TEST_COMPLETED
remote-operations	traceRoutePathChange	1.3.6.1.2.1.81.0	6	1	info	SNMP_TRAP_TRACE_ ROUTE_PATH_ CHANGE
remote-operations	traceRouteTestFailed	1.3.6.1.2.1.81.0	6	2	info	SNMP_TRAP_ TRACE_ROUTE_TEST _FAILED
remote-operations	traceRouteTestCompleted	1.3.6.1.2.1.81.0	6	3	info	SNMP_TRAP_TRACE_ ROUTE_TEST_ COMPLETED
rmon-alarm	fallingAlarm	1.3.6.1.2.1.16	6	2	—	—
rmon-alarm	risingAlarm	1.3.6.1.2.1.16	6	1	—	—
routing	bgpEstablished	1.3.6.1.2.1.15.7	6	1	—	—
routing	bgpBackwardTransition	1.3.6.1.2.1.15.7	6	2	—	—
routing	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2	6	1	—	—
routing	ospfNbrStateChange	1.3.6.1.2.1.14.16.2	6	2	—	—
routing	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2	6	3	—	—
routing	ospfIfConfigError	1.3.6.1.2.1.14.16.2	6	4	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2	6	5	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2	6	6	—	—
routing	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2	6	7	—	—
routing	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	8	—	—
routing	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	9	—	—
routing	ospfTxRetransmit	1.3.6.1.2.1.14.16.2	6	10	—	—
routing	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2	6	11	—	—
routing	ospfOriginateLsa	1.3.6.1.2.1.14.16.2	6	12	—	—
routing	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2	6	13	—	—
routing	ospfLsdbOverflow	1.3.6.1.2.1.14.16.2	6	14	—	—

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag
routing	ospfLsdbApproachingOverflow	1.3.6.1.2.1.14.16.2	6	15	—	—
routing	ospfIfStateChange	1.3.6.1.2.1.14.16.2	6	16	—	—
startup	coldStart	1.3.6.1.4.1.2636	0	0	critical	SNMPD_TRAP_ COLD_START
startup	warmStart	1.3.6.1.4.1.2636	1	0	error	SNMPD_TRAP_ WARM_START
vrrp	vrrpTrapNewMaster	1.3.6.1.2.1.68	6	1	warning	VRRPD_ NEWMASTER_TRAP
vrrp	vrrpTrapAuthFailure	1.3.6.1.2.1.68	6	2	warning	VRRPD_AUTH_ FAILURE_TRAP

SNMPv1 also supports the following standard traps:

- SNMP Version 1 Standard Traps on page 121
- SNMP Version 1 Ping Traps MIB on page 122
- SNMP Version 1 Traceroute Traps MIB on page 124
- SNMP Version 1 VRRP Traps MIB on page 125

SNMP Version 1 Standard Traps

The JUNOS software supports the standard SNMP version 1 traps, which are taken from RFC 1215, *Convention for defining traps for use with the SNMP*:

```
coldStart          TRAP-TYPE
  ENTERPRISE   snmp
  DESCRIPTION
  “A coldStart trap signifies that the sending protocol entity is reinitializing
  itself such that the agent's configuration or the protocol entity implementation
  may be altered.”
  ::= 0

warmStart          TRAP-TYPE
  ENTERPRISE   snmp
  DESCRIPTION
  “A warmStart trap signifies that the sending protocol entity is reinitializing
  itself such that neither the agent configuration nor the protocol entity
  implementation is altered.”
  ::= 1

linkDown           TRAP-TYPE
  ENTERPRISE snmp
  VARIABLES   { ifIndex }
  DESCRIPTION
  "A linkDown trap signifies that the sending protocol entity recognizes a failure
  in one of the communication links represented in the agent's configuration."
  ::= 2
```

```
linkUp              TRAP-TYPE
  ENTERPRISE   snmp
  VARIABLES    { ifIndex }
  DESCRIPTION
  "A linkUp trap signifies that the sending protocol entity recognizes that one of
  the communication links represented in the agent's configuration has come
  up."
  ::= 3

authenticationFailure TRAP-TYPE
  ENTERPRISE   snmp
  DESCRIPTION
  "An authenticationFailure trap signifies that the sending protocol entity is the
  addressee of a protocol message that is not properly authenticated. While
  implementations of the SNMP must be capable of generating this trap, they
  must also be capable of suppressing the emission of such traps via an
  implementation- specific mechanism."
  ::= 4

egpNeighborLoss TRAP-TYPE
  ENTERPRISE snmp
  VARIABLES   { egpNeighAddr }
  DESCRIPTION
  "An egpNeighborLoss trap signifies that an EGP neighbor for whom the sending
  protocol entity was an EGP peer has been marked down and the peer
  relationship no longer obtains."
  ::= 5
```

SNMP Version 1 Ping Traps MIB

The JUNOS software supports the SNMP traps from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*, converted to SNMPv1 format:

-definition of ping MIB traps

```
SNMP Version 1 Traceroute Traps MIB
  pingProbeFailed TRAP-TYPE
  ENTERPRISE pingMIB
  VARIABLES   {
    pingCtlTargetAddressType, pingCtlTargetAddress,
    pingResultsOperStatus, pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress, pingResultsMinRtt,
    pingResultsMaxRtt, pingResultsAverageRtt,
    pingResultsProbeResponses, pingResultsSentProbes,
    pingResultsRttSumOfSquares, pingResultsLastGoodProbe
  }
  STATUS                        mandatory
  DESCRIPTION
  "Generated when a probe failure is detected when the corresponding
  pingCtlTrapGeneration object is set to probeFailure(0) subject to the value of
  pingCtlTrapProbeFailureFilter. The object pingCtlTrapProbeFailureFilter can be
  used to specify the number of successive probe failures that are required
  before this notification can be generated."
::= 1
```

```
pingTestFailed                  TRAP-TYPE
  ENTERPRISE                    pingMIB
  VARIABLES   {
    pingCtlTargetAddressType, pingCtlTargetAddress,
    pingResultsOperStatus, pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress, pingResultsMinRtt,
    pingResultsMaxRtt, pingResultsAverageRtt,
    pingResultsProbeResponses, pingResultsSentProbes,
    pingResultsRttSumOfSquares, pingResultsLastGoodProbe
  }
  STATUS                        mandatory
  DESCRIPTION
  "Generated when a ping test is determined to have failed when the
  corresponding pingCtlTrapGeneration object is set to testFailure(1). In this
  instance pingCtlTrapTestFailureFilter should specify the number of probes in a
  test required to have failed in order to consider the test as failed."
::= 2

pingTestCompleted               TRAP-TYPE
  ENTERPRISE pingMIB
  VARIABLES   {
    pingCtlTargetAddressType, pingCtlTargetAddress,
    pingResultsOperStatus, pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress, pingResultsMinRtt,
    pingResultsMaxRtt, pingResultsAverageRtt,
    pingResultsProbeResponses, pingResultsSentProbes,
    pingResultsRttSumOfSquares, pingResultsLastGoodProbe
  }
  STATUS                        mandatory
  DESCRIPTION
    "Generated at the completion of a ping test when the
    corresponding pingCtlTrapGeneration object is set to
    testCompletion(4)."
::= 3
```

SNMP Version 1 Traceroute Traps MIB

The JUNOS software supports the SNMP traps from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*, converted to SNMPv1 format:

```
-definition of traceroute traps

    traceRoutePathChange  TRAP-TYPE
      ENTERPRISE          traceRouteMIB
      VARIABLES {
        traceRouteCtlTargetAddressType,
        traceRouteCtlTargetAddress,
        traceRouteResultsIpTgtAddrType,
        traceRouteResultsIpTgtAddr
      }
      STATUS              mandatory
      DESCRIPTION
        "The path to a target has changed."
    ::= 1

    traceRouteTestFailed  TRAP-TYPE
      ENTERPRISE          traceRouteMIB
      VARIABLES {
        traceRouteCtlTargetAddressType,
        traceRouteCtlTargetAddress,
        traceRouteResultsIpTgtAddrType,
        traceRouteResultsIpTgtAddr
      }
      STATUS              mandatory
      DESCRIPTION
      "Could not determine the path to a target."
      ::= 2

    traceRouteTestCompletedTRAP-TYPE
      ENTERPRISE          traceRouteMIB
      VARIABLES {
        traceRouteCtlTargetAddressType,
        traceRouteCtlTargetAddress,
        traceRouteResultsIpTgtAddrType,
        traceRouteResultsIpTgtAddr
      }
      STATUS              mandatory
      DESCRIPTION
        "The path to a target has just been determined."
    ::= 3
```

SNMP Version 1 VRRP Traps MIB

The JUNOS software supports the SNMP traps from RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*, converted to SNMPv1 format:

```
-definition of vrrp traps

      vrrpTrapNewMaster    TRAP-TYPE
        ENTERPRISE         vrrpMIB
        VARIABLES {
          vrrpOperMasterIpAddr
        }
-       STATUS             mandatory
        DESCRIPTION
        "The newMaster trap indicates that the sending agent has transitioned to
        'Master' state."
      ::= 1

      vrrpTrapAuthFailure  TRAP-TYPE
        ENTERPRISE         vrrpMIB
        VARIABLES {
          vrrpTrapPacketSrc
          vrrpTrapAuthErrorType
        }
-       STATUS             mandatory
        DESCRIPTION
        "A vrrpAuthFailure trap signifies that a packet has been received from a router
        whose authentication key or authentication type conflicts with this router's
        authentication key or authentication type. Implementation of this trap is
        optional."
      ::= 2
```

Standard SNMP Version 2 Traps

Table 13 provides an overview of the standard SNMPv2 traps supported by the JUNOS software. The traps are organized first by trap category and then by trap name and include their snmpTrapOID. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system log messages, see the *JUNOS System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS System Basics Configuration Guide*.

Table 13: Standard Supported SNMP Version 2 Traps

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag
authentication	authenticationFailure	1.3.6.1.6.3.1.1.5.5	notice	SNMPD_TRAP_GEN_FAILURE
link	linkDown	1.3.6.1.6.3.1.1.5.3	warning	SNMP_TRAP_LINK_DOWN
link	linkUp	1.3.6.1.6.3.1.1.5.4	info	SNMP_TRAP_LINK_UP
remote-operations	pingProbeFailed	1.3.6.1.2.1.80.0.1	info	SNMP_TRAP_PING_PROBE_FAILED
remote-operations	pingTestFailed	1.3.6.1.2.1.80.0.2	info	SNMP_TRAP_PING_TEST_FAILED
remote-operations	pingTestCompleted	1.3.6.1.2.1.80.0.3	info	SNMP_TRAP_PING_TEST_COMPLETED
remote-operations	traceRoutePathChange	1.3.6.1.2.1.81.0.1	info	SNMP_TRAP_TRACE_ROUTE_PATH_CHANGE
remote-operations	traceRouteTestFailed	1.3.6.1.2.1.81.0.2	info	SNMP_TRAP_TRACE_ROUTE_TEST_FAILED
remote-operations	traceRouteTestCompleted	1.3.6.1.2.1.81.0.3	info	SNMP_TRAP_TRACE_ROUTE_TEST_COMPLETED
rmon-alarm	fallingAlarm	1.3.6.1.2.1.15.7.1	—	—
rmon-alarm	risingAlarm	1.3.6.1.2.1.15.7.2	—	—
routing	bgpEstablished	1.3.6.1.2.1.15.7.1	—	—
routing	bgpBackwardTransition	1.3.6.1.2.1.15.7.2	—	—
routing	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2.1	—	—
routing	ospfNbrStateChange	1.3.6.1.2.1.14.16.2.2	—	—
routing	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2.3	—	—
routing	ospfIfConfigError	1.3.6.1.2.1.14.16.2.4	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.5	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.6	—	—
routing	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2.7	—	—
routing	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2.8	—	—
routing	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2.9	—	—
routing	ospfTxRetransmit	1.3.6.1.2.1.14.16.2.10	—	—

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag
routing	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2.11	—	—
routing	ospfOriginateLsa	1.3.6.1.2.1.14.16.2.12	—	—
routing	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2.13	—	—
routing	ospfLsdbOverflow	1.3.6.1.2.1.14.16.2.14	—	—
routing	ospfLsdbApproachingOverflow	1.3.6.1.2.1.14.16.2.15	—	—
routing	ospfIfStateChange	1.3.6.1.2.1.14.16.2.16	—	—
startup	coldStart	1.3.6.1.6.3.1.1.5.1	critical	SNMPD_TRAP_COLD_START
startup	warmStart	1.3.6.1.6.3.1.1.5.2	error	SNMPD_TRAP_WARM_START
vrrp	vrrpTrapNewMaster	1.3.6.1.2.1.68.0.1	warning	VRRPD_NEWMASTER_TRAP
vrrp	vrrpTrapAuthFailure	1.3.6.1.2.1.68.0.2	warning	VRRPD_AUTH_FAILURE_TRAP

The JUNOS software supports the following standard SNMP version 2 traps:

- SNMP Version 2 Standard Traps on page 128
- SNMP Version 2 BGP Traps MIB on page 129
- SNMP Version 2 OSPF Traps MIB on page 130
- SNMP Version 2 Ping Traps MIB on page 135
- SNMP Version 2 Traceroute Traps MIB on page 136
- SNMP Version 2 VRRP Traps MIB on page 137

SNMP Version 2 Standard Traps

The JUNOS software supports the standard SNMP version traps, which are taken from RFC 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*, and RFC 2863, *The Interfaces Group MIB*:

```
coldStart          NOTIFICATION-TYPE
   STATUS          current
   DESCRIPTION
   "A coldStart trap signifies that the SNMPv2 entity, acting in an agent role, is
   reinitializing itself and that its configuration may have been altered."
::= { snmpTraps 1 }

warmStart          NOTIFICATION-TYPE
   STATUS          current
   DESCRIPTION
   "A warmStart trap signifies that the SNMPv2 entity, acting in an agent role, is
   reinitializing itself such that its configuration is unaltered."
::= { snmpTraps 2 }

linkDown           NOTIFICATION-TYPE
   OBJECTS {
      ifIndex
      ifAdminStatus
      ifOperStatus
   }
   STATUS          current
   DESCRIPTION
   "A linkDown trap signifies that the SNMP entity, acting in an agent role, has
   detected that the ifOperStatus object for one of its communication links is about
   to enter the down state from some other state (but not from the notPresent
   state). This other state is indicated by the included value of ifOperStatus."
::= { snmpTraps 3 }

linkUp             NOTIFICATION-TYPE
   OBJECTS {
      ifIndex
      ifAdminStatus
      ifOperStatus
   }
   STATUS          current
   DESCRIPTION
   "A linkUp trap signifies that the SNMP entity, acting in an agent role, has
   detected that the ifOperStatus object for one of its communication links left the
   down state and transitioned into some other state (but not into the notPresent
   state). This other state is indicated by the included value of ifOperStatus."
::= { snmpTraps 4 }

authenticationFailureNOTIFICATION-TYPE
   STATUS          current
   DESCRIPTION
   "An authenticationFailure trap signifies that the SNMPv2 entity, acting in an
   agent role, has received a protocol message that is not properly authenticated.
   While all implementations of the SNMPv2 must be capable of generating
   this trap, the snmpEnableAuthenTraps object indicates whether this trap will be
   generated."
::= { snmpTraps 5 }
```

SNMP Version 2 BGP Traps MIB

The JUNOS software supports the Border Gateway Protocol (BGP) standard SNMP version 2 traps. The following descriptions are taken from RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*:

```
bgpEstablished      NOTIFICATION-TYPE
  OBJECTS  {
    bgpPeerLastError
    bgpPeerState
    }
  STATUS            current
  DESCRIPTION
  "The BGP Established event is generated when the BGP FSM enters the
  ESTABLISHED state."
::= { bgpTraps 1 }

bgpBackwardTransitionNOTIFICATION-TYPE
  OBJECTS  {
    bgpPeerLastError
    bgpPeerState
  }
  STATUS            current
  DESCRIPTION
  "The BGPBackwardTransition Event is generated when the BGP FSM moves from
  a higher numbered state to a lower numbered state."
::= { bgpTraps 2 }
```

SNMP Version 2 OSPF Traps MIB

The JUNOS software supports the Open Shortest Path First (OSPF) SNMP version 2 traps. The following descriptions are taken from RFC 1850, *OSPF Version 2 Management Information Base*:

```
ospfIfStateChange  NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfIfIpAddress,
      ospfAddressLessIf,
      ospfIfState } – The new state
   STATUS           current
DESCRIPTION
"An ospfIfStateChange trap signifies that there has been a change in the state of a
non-virtual OSPF interface. This trap should be generated when the interface state
regresses (e.g., goes from Dr to Down) or progresses to a terminal state (i.e.,
Point-to-Point, DR Other, Dr, or Backup)."
::= { ospfTraps 16 }

ospfVirtIfStateChange NOTIFICATION-TYPE
OBJECTS {
   ospfRouterId, – The originator of the trap
   ospfVirtIfAreaId,
   ospfVirtIfNeighbor,
   ospfVirtIfState } – The new state
STATUS           current
DESCRIPTION
"An ospfIfStateChange trap signifies that there has been a change in the state of
an OSPF virtual interface. This trap should be generated when the interface state
regresses (e.g., goes from Point-to-Point to Down) or progresses to a terminal
state (i.e., Point)."
::= { ospfTraps 1 }

ospfNbrStateChange NOTIFICATION-TYPE
OBJECTS {
   ospfRouterId, – The originator of the trap
   ospfNbrIpAddr,
   ospfNbrAddressLessIndex,
   ospfNbrRtrId,
   ospfNbrState
   } – The new state
   STATUS           current
   DESCRIPTION
   "An ospfNbrStateChange trap signifies that there has been a change in the state
   of a non-virtual OSPF neighbor. This trap should be generated when the neighbor
   state regresses (e.g., goes from Attempt or Full to 1-Way or Down) or progresses
   to a terminal state (e.g., 2-Way or Full). When a neighbor transitions from or to
   Full on non-broadcast multi-access and broadcast networks, the trap should be
   generated by the designated router. A designated router transitioning to Down
   will be noted by ospfIfStateChange."
::= { ospfTraps 2 }
```

```
ospfVirtNbrStateChange NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfVirtNbrArea,
      ospfVirtNbrRtrId,
      ospfVirtNbrState
   } – The new state
   STATUS          current
   DESCRIPTION
    "An ospfIfStateChange trap signifies that there has been a change in the state
   of an OSPF virtual neighbor. This trap should be generated when the neighbor
   state regresses(e.g., goes from Attempt or Full to 1-Way or Down) or progresses
   to a terminal state (e.g., Full)."
::= { ospfTraps 3 }

ospfIfConfigError NOTIFICATION-TYPE
OBJECTS {
   ospfRouterId, – The originator of the trap
   ospfIfIpAddress,
   ospfAddressLessIf,
   ospfPacketSrc, – The source IP address
   ospfConfigErrorType, – Type of error
   ospfPacketType
}
STATUS          current
DESCRIPTION
   "An ospfIfConfigError trap signifies that a packet has been received on a
   non-virtual interface from a router whose configuration parameters conflict with
   this router's configuration parameters. Note that the event optionMismatch
   should cause a trap only if it prevents an adjacency from forming."
::= { ospfTraps 4 }

ospfVirtIfConfigError NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfVirtIfAreaId,
      ospfVirtIfNeighbor,
      ospfConfigErrorType, – Type of error
      ospfPacketType
   }
   STATUS          current
   DESCRIPTION
   "An ospfConfigError trap signifies that a packet has been received on a virtual
   interface from a router whose configuration parameters conflict with this router's
   configuration parameters. Note that the event optionMismatch should cause a
   trap only if it prevents an adjacency from forming."
::= { ospfTraps 5 }
```

```
ospfIfAuthFailure NOTIFICATION-TYPE
   OBJECTS {ospfRouterId, – The originator of the trap
      ospfIfIpAddress,
      ospfAddressLessIf,
      ospfPacketSrc, – The source IP address
      ospfConfigErrorType, – authTypeMismatch or
                                 – authFailure
      ospfPacketType
}
STATUS          current
DESCRIPTION
"An ospfIfAuthFailure trap signifies that a packet has been received on a
non-virtual interface from a router whose authentication key or authentication type
conflicts with this router's authentication key or authentication type."
::= { ospfTraps 6 }

ospfVirtIfAuthFailure NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfVirtIfAreaId,
      ospfVirtIfNeighbor,
      ospfConfigErrorType, – authTypeMismatch or
                               – authFailure
      ospfPacketType }
   STATUS          current
   DESCRIPTION
   "An ospfVirtIfAuthFailure trap signifies that a packet has been received on a
   virtual interface from a router whose authentication key or authentication type
   conflicts with this router's authentication key or authentication type."
::= { ospfTraps 7 }

ospfIfRxBadPacket NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfIfIpAddress,
      ospfAddressLessIf,
      ospfPacketSrc, – The source IP address
      ospfPacketType
   }
   STATUS          current
   DESCRIPTION
   "An ospfIfRxBadPacket trap signifies that an OSPF packet has been received on
   a nonvirtual interface that cannot be parsed."
::= { ospfTraps 8 }
```

```
ospfVirtIfRxBadPacket NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfVirtIfAreaId,
      ospfVirtIfNeighbor,
      ospfPacketType
   }
   STATUS          current
   DESCRIPTION
   "An ospfRxBadPacket trap signifies that an OSPF packet has been received on a
   virtual interface that cannot be parsed."
::= { ospfTraps 9 }

ospfTxRetransmit NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfIfIpAddress,
      ospfAddressLessIf,
      ospfNbrRtrId, – Destination
      ospfPacketType,
      ospfLsdbType,
      ospfLsdbLsid,
      ospfLsdbRouterId
}
STATUS          current
DESCRIPTION
"An ospfTxRetransmit trap signifies that an OSPF packet has been retransmitted
on a nonvirtual interface. All packets that may be re-transmitted are associated
with an LSDB entry. The LS type, LS ID, and Router ID are used to identify the
LSDB entry."
::= { ospfTraps 10 }

ospfVirtIfTxRetransmit NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfVirtIfAreaId,
      ospfVirtIfNeighbor,
      ospfPacketType,
      ospfLsdbType,
      ospfLsdbLsid,
      ospfLsdbRouterId
   }
   STATUS          current
   DESCRIPTION
   "An ospfTxRetransmit trap signifies that an OSPF packet has been retransmitted
   on a virtual interface. All packets that may be retransmitted are associated with
   an LSDB entry. The LS type, LS ID, and Router ID are used to identify the LSDB
   entry."
::= { ospfTraps 11 }
```

```
ospfOriginateLsa NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfLsdbAreaId, – 0.0.0.0 for AS Externals
      ospfLsdbType,
      ospfLsdbLsid,
      ospfLsdbRouterId
   }
   STATUS          current
   DESCRIPTION
   "An ospfOriginateLsa trap signifies that a new LSA has been originated by this
   router. This trap should not be invoked for simple refreshes of LSAs (which
   happens every 30 minutes), but instead will only be invoked when an LSA is
   (re)originated due to a topology change. Additionally, this trap does not include
   LSAs that are being flushed because they have reached MaxAge."
::= { ospfTraps 12 }

ospfMaxAgeLsa NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfLsdbAreaId, – 0.0.0.0 for AS Externals
      ospfLsdbType,
      ospfLsdbLsid,
      ospfLsdbRouterId
   }
   STATUS          current
   DESCRIPTION
   "An ospfMaxAgeLsa trap signifies that one of the LSAs in the router's link-state
   database has aged to MaxAge."
::= { ospfTraps 13 }

ospfLsdbOverflow NOTIFICATION-TYPE
   OBJECTS {
      ospfRouterId, – The originator of the trap
      ospfExtLsdbLimit
   }
   STATUS          current
   DESCRIPTION
   "An ospfLsdbOverflow trap signifies that the number of LSAs in the router's
   link-state database has exceeded ospfExtLsdbLimit."
::= { ospfTraps 14 }

ospfLsdbApproachingOverflow NOTIFICATION-TYPE
   OBJECTS {
         ospfRouterId, – The originator of the trap
         ospfExtLsdbLimit
   }
   STATUS          current
   DESCRIPTION
      "An ospfLsdbApproachingOverflow trap signifies that the number of LSAs in
      the router's link-state database has exceeded ninety percent of
      ospfExtLsdbLimit."
   := { ospfTraps 15 }
```

SNMP Version 2 Ping Traps MIB

The following descriptions for the SNMPv2 ping traps are from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*:

```
pingProbeFailed      NOTIFICATION-TYPE
  OBJECTS {
    pingCtlTargetAddressType,
    pingCtlTargetAddress,
    pingResultsOperStatus,
    pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress,
    pingResultsMinRtt,
    pingResultsMaxRtt,
    pingResultsAverageRtt,
    pingResultsProbeResponses,
    pingResultsSentProbes,
    pingResultsRttSumOfSquares,
    pingResultsLastGoodProbe
  }
  STATUS             current
  DESCRIPTION
  "Generated when a probe failure is detected when the corresponding
  pingCtlTrapGeneration object is set to probeFailure(0) subject to the value of
  pingCtlTrapProbeFailureFilter. The object pingCtlTrapProbeFailureFilter can be
  used to specify the number of successive probe failures that are required
  before this notification can be generated."
::= { pingNotifications 1 }

pingTestFailed       NOTIFICATION-TYPE
  OBJECTS {
    pingCtlTargetAddressType,
    pingCtlTargetAddress,
    pingResultsOperStatus,
    pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress,
    pingResultsMinRtt,
    pingResultsMaxRtt,
    pingResultsAverageRtt,
    pingResultsProbeResponses,
    pingResultsSentProbes,
    pingResultsRttSumOfSquares,
    pingResultsLastGoodProbe
  }
  STATUS               current
  DESCRIPTION
    "Generated when a ping test is determined to have failed when the
    corresponding pingCtlTrapGeneration object is set to testFailure(1). In this
    instance pingCtlTrapTestFailureFilter should specify the number of probes in a
    test required to have failed in order to consider the test as failed."
::= { pingNotifications 2 }
```

```
pingTestCompleted NOTIFICATION-TYPE
   OBJECTS {
      pingCtlTargetAddressType,
      pingCtlTargetAddress,
      pingResultsOperStatus,
      pingResultsIpTargetAddressType,
      pingResultsIpTargetAddress,
      pingResultsMinRtt,
      pingResultsMaxRtt,
      pingResultsAverageRtt,
      pingResultsProbeResponses,
      pingResultsSentProbes,
      pingResultsRttSumOfSquares,
      pingResultsLastGoodProbe
   }
   STATUS             current
   DESCRIPTION
      "Generated at the completion of a ping test when the corresponding
      pingCtlTrapGeneration object is set to testCompletion(4)."
::= { pingNotifications 3 }
```

SNMP Version 2 Traceroute Traps MIB

The following descriptions for the SNMPv2 traceroute traps are from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*:

```
traceRoutePathChange  NOTIFICATION-TYPE
   OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
   }
   STATUS                current
   DESCRIPTION
   "The path to a target has changed."
::= { traceRouteNotifications 1 }

traceRouteTestFailed     NOTIFICATION-TYPE
OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
   }
   STATUS                current
   DESCRIPTION
   "Could not determine the path to a target."
::= { traceRouteNotifications 2 }
```

```
traceRouteTestCompletedNOTIFICATION-TYPE
   OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
   }
   STATUS              current
   DESCRIPTION
   "The path to a target has just been determined."
::= { traceRouteNotifications 3 }
```

SNMP Version 2 VRRP Traps MIB

The following descriptions for the SNMPv2 Virtual Router Redundancy Protocol (VRRP) traps are from RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*:

```
-- vrrp trap definitions

vrrpTrapPacketSrc       OBJECT-TYPE
   SYNTAX               IpAddress
   MAX-ACCESS           accessible-for-notify
   STATUS               current
   DESCRIPTION
   "The IP address of an inbound VRRP packet. Used by
   vrrpTrapAuthFailure trap."
::= { vrrpOperations 5 }

vrrpTrapAuthErrorType   OBJECT-TYPE
   SYNTAX               INTEGER {
                        invalidAuthType (1),
                        authTypeMismatch (2),
                        authFailure (3)
                        }
   MAX-ACCESS           accessible-for-notify
   STATUS               current
   DESCRIPTION
   "Potential types of configuration conflicts. Used by vrrpAuthFailure trap."
```

Chapter 12

Summary of SNMP Configuration Statements

The following sections explain each of the Simple Network Management Protocol (SNMP) configuration statements. The statements are organized alphabetically.

agent-address

Syntax agent-address outgoing-interface;

Hierarchy Level [edit snmp trap-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Set the agent address of all SNMPv1 traps generated by this router. Currently, the only option is outgoing-interface, which sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.

Options outgoing-interface—Value of agent address of all SNMPv1 traps generated by this router. The outgoing-interface option sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.

Default: disabled (The agent address is not specified in SNMPv1 traps.)

Usage Guidelines See "Configuring the Agent Address for SNMP Traps" on page 35.

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

authorization

Syntax authorization *authorization*;

Hierarchy Level [edit snmp community *community-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Set the access authorization for SNMP Get, GetBulk, GetNext, and Set requests.

Options *authorization*—Access authorization level:

- read-only—Enable Get, GetNext, and GetBulk requests.
- read-write—Enable all requests, including Set requests. You must configure a view to enable Set requests.

Default: read-only

Usage Guidelines See "Configuring the SNMP Community String" on page 30.

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

categories

Syntax categories [*categories*];

Hierarchy Level [edit snmp trap-group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the types of traps that will be sent to the targets of the named trap group.

Default If you omit the categories statement, all trap types are included in trap notifications.

Options *categories*—One or more trap types.
Values: authentication, chassis, configuration, link, remote-operations, rmon-alarm, routing, sonet-alarms, startup, vrrp-events

Usage Guidelines See "Configuring SNMP Trap Groups" on page 35.

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

clients

Syntax
```
clients {
    address restrict;
}
```

Hierarchy Level [edit snmp community *community-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the IPv4 or IPv6 addresses of the SNMP client hosts that are authorized to use this community.

Default If you omit the clients statement, all SNMP clients using this community string are authorized to access the router.

Options *address*—Address of an SNMP client that is authorized to access this router. You must specify an address, not a hostname. To specify more than one client, include multiple *address* options.

restrict—(Optional) Do not allow the specified SNMP client to access the router.
Default: If you omit the restrict option after the address, access is permitted for this particular client.

Usage Guidelines See "Configuring the SNMP Community String" on page 30.

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

commit-delay

Syntax commit-delay *seconds*;

Hierarchy Level [edit snmp nonvolatile]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the timer for the SNMP set reply and start of the commit.

Default: 5 seconds

Usage Guidelines See "Configuring the Commit Delay Timer" on page 29.

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

community

Syntax
```
community community-name {
    authorization authorization;
    clients {
      address restrict;
    }
    view view-name;
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define an SNMP community. An SNMP community authorizes SNMP clients based on the source IP address of incoming SNMP request packets. A community also defines which MIB objects are available and the operations (read-only or read-write) allowed on those objects.

The SNMP client application specifies an SNMP community name in Get, GetBulk, GetNext, and Set SNMP requests.

Default If you omit the community statement, all SNMP requests are denied.

Options *community-name*—Community string. If the name includes spaces, enclose it in quotation marks (" ").

The remaining statements are explained separately.

Usage Guidelines See "Configuring the SNMP Community String" on page 30.

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

contact

Syntax
```
contact contact;
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the value of the MIB II sysContact object, which is the contact person for the managed system.

Options *contact*—Name of contact person. If the name includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See "Configuring the System Contact" on page 27.

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

description

Syntax description *description*;

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the value of the MIB II sysDescription object, which is the description of the system being managed.

Options *description*—System description. If the name includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See "Configuring the System Description" on page 28.

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

destination-port

Syntax destination-port <*port-number*>;

Hierarchy Level [edit snmp trap-group]

Release Information Statement introduced before JUNOS Release 7.4.

Description Assign a trap port number other than the default.

Options *port-number*—(Optional) SNMP trap port number.
Default: port 162

Usage Guidelines See "Configuring SNMP Trap Groups" on page 35.

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

engine-id

See engine-id on page 157

filter-duplicates

Syntax filter-duplicates;

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Filter duplicate Get, GetNext, or GetBulk SNMP requests.

Usage Guidelines See "Filtering Duplicate SNMP Requests" on page 28.

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

interface

Syntax interface [*interface-names*];

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the interfaces on which SNMP requests can be accepted.

Default If you omit this statement, SNMP requests entering the router through any interface will be accepted.

Options *interface-names*—Names of one or more logical interfaces.

Usage Guidelines See "Configuring the Interfaces on Which SNMP Requests Can Be Accepted" on page 38.

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

location

Syntax location *location*;

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the value of the MIB II sysLocation object, which is the physical location of the managed system.

Options *location*—Location of the local system. You must enclose the name within quotation marks (" ").

Usage Guidelines See "Configuring the System Location" on page 27.

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

name

Syntax name *name*;

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Set the system name from the command-line interface.

Options *name*—System name override.

Usage Guidelines See "Configuring the System Name" on page 29.

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

nonvolatile

Syntax
```
nonvolatile {
    commit-delay seconds;
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure options for SNMP set requests.

The statement is explained separately in this chapter.

Usage Guidelines See "Configuring the Commit Delay Timer" on page 29.

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

oid

Syntax oid *object-identifier* (include | exclude);

Hierarchy Level [edit snmp view *view-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify an object identifier (OID) used to represent a subtree of MIB objects.

Options *object-identifier*—OID used to represent a subtree of MIB objects. All MIB objects represented by this statement have the specified OID as a prefix. It can be specified either by a sequence of dotted integers or by a subtree name.

include—Include the subtree of MIB objects represented by the specified OID.

exclude—Exclude the subtree of MIB objects represented by the specified OID.

Usage Guidelines See "Configuring MIB Views" on page 39.

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

snmp

Syntax snmp { ... }

Hierarchy Level [edit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure SNMP.

Usage Guidelines See "Configuring SNMP" on page 25.

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

source-address

Syntax source-address *address*;

Hierarchy Level [edit snmp trap-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Set the source address of every SNMP trap packet sent by this router to a single address regardless of the outgoing interface. If the source address is not specified, the default is to use the address of the outgoing interface as the source address.

Options *address*—Source address of SNMP traps. You can configure the source address of trap packets two ways: lo0 or a valid IPv4 address configured on one of the router interfaces. The value lo0 indicates that the source address of all SNMP trap packets will be set to the lowest loopback address configured at interface lo0.

Default: disabled (The source address is the address of outgoing interface.)

Usage Guidelines See "Configuring the Source Address for SNMP Traps" on page 33.

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

targets

Syntax
```
targets {
    address;
}
```

Hierarchy Level [edit snmp trap-group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure one or more systems to receive SNMP traps.

Options *address*—IPv4 or IPv6 address of the system to receive traps. You must specify an address, not a hostname.

Usage Guidelines See "Configuring SNMP Trap Groups" on page 35.

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

traceoptions

Syntax
```
traceoptions {
    file size size files number;
    flag flag;
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description The output of the tracing operations is placed into log files in the /var/log directory. Each log file is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:

- chassisd
- craftd
- ilmid
- mib2d
- rmopd
- serviced
- snmpd

Options

files *number*—(Optional) Maximum number of trace files per SNMP subagent. When a trace file (for example, snmpd) reaches its maximum size, it is archived by being renamed to snmpd.0. The previous snmpd.1 is renamed to snmpd.2, and so on. The oldest archived file is deleted.

Range: 2 through 1000 files
Default: 10 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements:

- all—Trace all SNMP events
- general—Trace general events
- interface-stats—Trace physical and logical interface statistics
- nonvolatile-sets—Trace nonvolatile SNMP set request handling
- pdu—Trace SNMP request and response packets
- protocol-timeouts—Trace SNMP response timeouts
- routing-socket—Trace routing socket calls
- subagent—Trace subagent restarts
- timer—Trace internal timer events
- varbind-error—Trace variable binding errors

size *size*—(Optional) Maximum size, in kilobytes (KB), of each trace file before it is closed and archived.

Range: 1 KB through the maximum file size supported on your system
Default: 1000 KB

Usage Guidelines

See "Tracing SNMP Activity" on page 40.

Required Privilege Level

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

trap-group

Syntax

```
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
      address;
    }
    version (all | v1 | v2);
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a named group of hosts to receive the specified trap notifications. The name of the trap group is embedded in SNMP trap notification packets as one variable binding (varbind) known as the community name. At least one trap group must be configured for SNMP traps to be sent.

Options *group-name*—Name of the trap group. If the name includes spaces, enclose it in quotation marks (" ").

The remaining statements are explained separately.

Usage Guidelines See “Configuring SNMP Trap Groups” on page 35.

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

trap-options

Syntax

```
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router to a single address, regardless of the outgoing interface. In addition, you can set the agent address of each SNMPv1 trap. For more information on the contents of SNMPv1 traps, see RFC 1157.

Options The remaining statements are explained separately.

Default: disabled

Usage Guidelines See “Configuring SNMP Trap Groups” on page 35.

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

version

Syntax version (all | v1 | v2);

Hierarchy Level [edit snmp trap-group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the version number of SNMP traps.

Options all—Send an SNMPv1 and SNMPv2 trap for every trap condition.

v1—Send SNMPv1 traps only.

v2—Send SNMPv2 traps only.

Default: all

Usage Guidelines See "Configuring SNMP Trap Groups" on page 35.

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

view

See the following sections:

- view (Associating MIB View with a Community) on page 151
- view (Configuring MIB View) on page 152

view (Associating MIB View with a Community)

Syntax view *view-name*;

Hierarchy Level [edit snmp community *community-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate a view with a community. A view represents a group of MIB objects.

Options *view-name*—Name of the view. You must use a view name already configured in the view statement at the [edit snmp] hierarchy level.

Usage Guidelines See "Configuring the SNMP Community String" on page 30.

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

view (Configuring MIB View)

Syntax
```
view view-name {
    oid object-identifier (include | exclude);
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define a MIB view. A MIB view identifies a group of MIB objects. Each MIB object in a view has a common OID prefix. Each object identifier represents a subtree of the MIB object hierarchy. The view statement uses a view to specify a group of MIB objects on which to define access. To enable a view, you must associate the view with a community by including the view statement at the [edit snmp community *community-name*] hierarchy level.

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

Options *view-name*—Name of the view

The remaining statements are explained separately.

Usage Guidelines See “Configuring MIB Views” on page 39.

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

See Also community on page 142

Chapter 13

Summary of SNMPv3 Configuration Statements

The following sections explain each of the SNMPv3 configuration statements. The statements are organized alphabetically.

address

Syntax address *address*;

Hierarchy Level [edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the SNMP target address.

Options *address*—IPv4 address of the system to receive traps or informs. You must specify an address, not a hostname.

Usage Guidelines See "Configuring the Address" on page 66.

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

address-mask

Syntax address-mask *address-mask*;

Hierarchy Level [edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Verify the source addresses for a group of target addresses.

Options *address-mask* combined with the address defines a range of addresses.

Usage Guidelines See "Configuring the Address Mask" on page 66.

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

authentication-md5

Syntax
```
authentication-md5 {
    authentication-password authentication-password;
}
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the MD5 as the authentication type for the SNMPv3 user.

Options *authentication-password*—Password that generates the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

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

Usage Guidelines See "Configuring the MD5 Authentication" on page 50.

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

authentication-none

Syntax
```
authentication-none;
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure no authentication for the SNMPv3 user.

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

Usage Guidelines See "Configuring No Authentication" on page 51.

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

authentication-password

Syntax authentication-password *authentication-password*;

Hierarchy Level [edit snmp v3 usm local-engine user *username* authentication-md5],
[edit snmp v3 usm local-engine user *username* authentication-sha]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure password for authentication.

Options *authentication-password*—Password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See "Configuring the MD5 Authentication" on page 50 and "Configuring the SHA Authentication" on page 51.

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

authentication-sha

Syntax
```
authentication-sha {
    authentication-password authentication-password;
}
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the SHA as the authentication type for the SNMPv3 user

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

Options *authentication-password*—The password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See “Configuring the SHA Authentication” on page 51.

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

community-name

Syntax community-name *community-name*;

Hierarchy Level [edit snmp v3 snmp-community *community-index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The community name defines an SNMP community. The SNMP community authorizes SNMPv1 or SNMPv2 clients. The access privileges associated with the configured security name define which MIB objects are available and the operations (notify, read, or write) allowed on those objects

Options *community-name*—A community string for an SNMPv1 or SNMPv2c community. If unconfigured, it is the same as the community index. If the name includes spaces, enclose it in quotation marks (" ").

NOTE: Community names must be unique. You cannot configure the same community name at the [edit snmp community] and [edit snmp v3 snmp-community *community-index*] hierarchy levels.

The community name at the [edit snmp v3 snmp-community *community-index*] hierarchy level is encrypted and not displayed in the CLI.

Usage Guidelines See “Configuring the SNMP Community” on page 77.

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

engine-id

Syntax

```
engine-id {
    (local engine-id-suffix | use-default-ip-address | use-mac-address);
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description The local engine ID is defined as the administratively unique identifier of an SNMPv3 engine, and is used for identification, not for addressing. There are two parts of an engine ID: prefix and suffix. The prefix is formatted according to the specifications defined in RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*. You can configure the suffix here.

NOTE: SNMPv3 authentication and encryption keys are generated based on the associated passwords and the engine ID. If you configure or change the engine ID, you must commit the new engine ID before you configure SNMPv3 users. Otherwise the keys generated from the configured passwords will be based on the previous engine ID.

For the engine ID, we recommend using the MAC address of fxp0.

Options

local *engine-id-suffix*—The engine ID suffix is explicitly configured.

use-default-ip-address—The engine ID suffix is generated from the default IP address.

use-mac-address—The SNMP engine identifier is generated from the MAC address of the management interface on the routing platform.

Default: use-default-ip-address

Usage Guidelines See "Configuring the Local Engine ID" on page 48.

Required Privilege Level

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

group

See the following sections:

group (Configuring)

Syntax group *group-name*;

Hierarchy Level [edit snmp v3 vacm access]

Release Information Statement introduced before JUNOS Release 7.4.

Description Assign the security name to a group.

Options *group-name*—SNMPv3 group name created for the SNMPv3 group.

Usage Guidelines See "Configuring the Group" on page 57.

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

group (Defining Access Privileges for an SNMPv3 Group)

Syntax group *group-name*;

Hierarchy Level [edit snmp v3 vacm security-to-group security-model (usm | v1 | v2) security-name *security-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define access privileges granted to a group.

Options *group-name*—Identifies a collection of SNMP security names that belong to the same access policy SNMP.

Usage Guidelines See "Configuring the Group" on page 61.

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

inform-retry-count

Syntax inform-retry-count *number*;

Hierarchy Level [edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure the retry count for SNMP informs.

Options *number*—Maximum number of times the inform is transmitted if no acknowledgement is received. If no acknowledgement is received after the inform is transmitted the maximum number of times, the inform message is discarded.
Default: 3

Usage Guidelines See "Configuring SNMP Informs" on page 72.

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

See Also inform-timeout on page 159.

inform-timeout

Syntax inform-timeout *seconds*;

Hierarchy Level [edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure the timeout period (in seconds) for SNMP informs.

Options *seconds*—Number of seconds to wait for an inform acknowledgement. If no acknowledgement is received within the timeout period, the inform is retransmitted.
Default: 15

Usage Guidelines See "Configuring SNMP Informs" on page 72.

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

See Also inform-retry-count on page 159.

local-engine

Syntax

```
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 {
            privacy-password privacy-password;
        }
    }
}
```

Hierarchy Level [edit snmp v3 usm]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure local-engine information for the user-based security model (USM).

The remaining statements are explained separately.

Usage Guidelines See “Creating SNMPv3 Users” on page 49.

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

message-processing-model

Syntax message-process-model (v1 | v2c | v3);

Hierarchy Level [edit snmp v3 target-parameters *target-parameter-name* parameters]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the message processing model to be used when generating SNMP notifications.

Options v1—SNMPv1 message process model.

v2c—SNMPv2c message process model.

v3—SNMPv3 message process model.

Usage Guidelines See "Configuring the Message Processing Model" on page 69.

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

notify

Syntax
```
notify name {
    tag tag-name;
    type (trap | inform);
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.
type *inform* option added in JUNOS Release 7.4.

Description Select management targets for notifications as well as the type of notifications. Notifications can be either traps or informs.

Options *name*—Name assigned to the notification.

tag-name—Notifications are sent to all targets configured with this tag.

type—Notification type is trap or inform. Traps are unconfirmed notifications. Informs are confirmed notifications.

Usage Guidelines See "Configuring the Trap Target Address" on page 65 and "Configuring the Inform Notification Type and Target Address" on page 74.

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

notify-filter

See the following sections:

notify-filter (Applying to Management Target)

Syntax notify-filter *profile-name*;

Hierarchy Level [edit snmp v3 target-parameters *target-parameters-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the notify filter to use by a specific set of target parameters.

Options *profile-name*—Name of the notify filter to apply to notifications.

Usage Guidelines See "Applying the Trap Notification Filter" on page 69.

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

notify-filter (Configuring)

Syntax
```
notify-filter profile-name {
    oid oid (include | exclude);
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define a group of MIB objects on which to define access. The notify filter limits the type of traps or informs sent to the NMS.

Options *profile-name*—Name assigned to the notify filter.

The remaining statement is explained separately.

Usage Guidelines See "Configuring the Trap Notification Filter" on page 64.

See Also oid on page 163.

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

notify-view

Syntax notify-view *view-name*;

Hierarchy Level [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate the view with a community or a group name (SNMPv3).

Options *view-name*—Name of the view to which the SNMP user group has access.

Usage Guidelines See "Configuring the Notify View" on page 58.

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

See Also See "Configuring MIB Views" on page 54.

oid

Syntax oid *oid* (include | exclude);

Hierarchy Level [edit snmp v3 notify-filter *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify an object identifier (OID) used to represent a subtree of MIB objects.

Options *oid*—Object identifier used to represent a subtree of MIB objects. All MIB objects represented by this statement have the specified OID as a prefix. It can be specified either by a sequence of dotted integers or by a subtree name.

include—Include the subtree of MIB objects represented by the specified OID.

exclude—Exclude the subtree of MIB objects represented by the specified OID.

Usage Guidelines See "Configuring the Trap Notification Filter" on page 64.

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

parameters

Syntax
```
parameters {
    message-processing-model (v1 | v2c | v3);
    security-model (usm | v1 | v2c);
    security-level (none | authentication | privacy);
    security-name security-name;
}
```

Hierarchy Level [edit snmp v3 target-parameters *target-parameters-name]*

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a set of target parameters.

The remaining statements are explained separately.

Usage Guidelines See "Defining the Trap Target Parameters" on page 68.

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

port

Syntax port <*port-number*>;

Hierarchy Level [edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a UDP port number for an SNMP target.

Options <*port-number*>—(Optional) Port number for an SNMP target.
Default: port number 162

Usage Guidelines See "Configuring the Port" on page 66.

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

privacy-3des

Syntax
```
privacy-3des {
    privacy-password privacy-password;
}
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the triple Data Encryption Standard (3DES) for the SNMPv3 user.

Options *privacy-password*—The password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See "Configuring the Encryption Type" on page 51.

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

privacy-aes128

Syntax
```
privacy-aes128 {
    privacy-password privacy-password;
}
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the Advanced Encryption Standard encryption algorithm (CFB128-AES-128 Privacy Protocol) for the SNMPv3 user.

Options *privacy-password*—The password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See "Configuring the Encryption Type" on page 51.

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

privacy-des

Syntax
```
privacy-des {
    privacy-password privacy-password;
}
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure Data Encryption Standard (DES) for the SNMPv3 user.

Options *privacy-password*—The password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See "Configuring the Encryption Type" on page 51.

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

privacy-none

Syntax
```
privacy-none;
```

Hierarchy Level [edit snmp v3 usm local-engine user *username*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure no encryption for the SNMPv3 user.

Usage Guidelines See "Configuring the Encryption Type" on page 51.

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

privacy-password

Syntax privacy-password *privacy-password*;

Hierarchy Level [edit snmp v3 usm local-engine user *username* privacy-3des],
[edit snmp v3 usm local-engine user *username* privacy-aes128],
[edit snmp v3 usm local-engine user *username* privacy-des]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a privacy password for the SNMPv3 user.

Options *privacy-password*—The password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

- The password must be at least 8 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

Usage Guidelines See "Configuring the Encryption Type" on page 51.

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

read-view

Syntax read-view *view-name*;

Hierarchy Level [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate the view with a community or a group name (SNMPv3).

Options *view-name*—The name of the view to which the SNMP user group has access.

Usage Guidelines See "Configuring the Read View" on page 59.

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

See Also See "Configuring MIB Views" on page 54.

remote-engine

Syntax

```
remote-engine engine-id {
    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 {
            privacy-password privacy-password;
        }
    }
}
```

Hierarchy Level [edit snmp v3 usm]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure remote engine information for the user-based security model (USM). To send inform messages to an SNMPv3 user on a remote device, you must configure the engine identifier for the SNMP agent on the remote device where the user resides.

The remaining statements are explained separately.

Options *engine-id*—Engine identifier. Used to compute the security digest for authenticating and encrypting packets sent to a user on the remote host.

Usage Guidelines See “Configuring the Remote Engine and Remote User” on page 73.

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

security-level

See the following sections:

security-level (Defining Access Privileges)

Syntax security-level (none | authentication | privacy);

Hierarchy Level [edit snmp v3 vacm access group group-name default-context-prefix security-model (any | usm | v1 | v2c)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the security level used for access privileges.

Options none—No authentication and no encryption.

authentication—Provides authentication but no encryption.

privacy—Provides authentication and encryption.

Default: none

Usage Guidelines See "Configuring the Security Level" on page 57.

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

security-level (Generating SNMP Notifications)

Syntax security-level (none | authentication | privacy);

Hierarchy Level [edit snmp v3 target-parameters *target-parameters-name* parameters]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the security level to use when generating SNMP notifications.

Options none—No authentication and no encryption.

authentication—Provides authentication but no encryption.

privacy—Provides authentication and encryption.
Default: none

Usage Guidelines See "Configuring the Security Level" on page 70.

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

security-model

See the following sections:

security-model (Access Privileges)

Syntax security-model (usm | v1 | v2c);

Hierarchy Level [edit snmp v3 vacm access group *group-name* default-context-prefix]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a group's security model used for access privileges.

Options usm—SNMPv3 security model.

v1—SNMPv1 security model.

v2c—SNMPv2c security model.

Usage Guidelines See "Configuring the Security Model" on page 57.

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

security-model (Group)

Syntax security-model (usm | v1 | v2c);

Hierarchy Level [edit snmp v3 vacm security-to-group]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define a security model for a group.

Options usm—SNMPv3 security model.

v1—SNMPv1 security model.

v2c—SNMPv2c security model.

Usage Guidelines See "Configuring the Security Model" on page 60.

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

security-model (SNMP Notifications)

Syntax security-model (usm | v1 | v2c);

Hierarchy Level [edit snmp v3 target-parameters *target-parameters-name* parameters]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a group's security model used with sending notifications.

Options usm—SNMPv3 security model.

v1—SNMPv1 security model.

v2c—SNMPv2c security model.

Usage Guidelines See "Configuring the Security Model" on page 70.

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

security-name

See the following sections:

- security-name (Community String) on page 171
- security-name (Security Group) on page 172
- security-name (SNMP Notifications) on page 172

security-name (Community String)

Syntax security-name *security-name*;

Hierarchy Level [edit snmp v3 snmp-community *community-index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate the community string configured at the [edit snmp v3 snmp-community *community-index*] hierarchy level to a security name.

Options *security-name* used when performing access control.

NOTE: The security name must match the configured security name at the [edit snmp v3 target-parameters *target-parameters-name* parameters] hierarchy level when configuring traps or informs.

Usage Guidelines See "Configuring the Security Names" on page 78.

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

security-name (Security Group)

Syntax security-name *security-name*;

Hierarchy Level [edit snmp v3 vacm security-to-group security-model (usm | v1 |v2c)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate a group or a community string with a configured security group.

Options *security-name*—The username configured at the [edit snmp v3 usm local-engine user *username*] hierarchy level. For SNMPv1 and SNMPv2c, the security name is the community string configured at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

Usage Guidelines See "Configuring the Security Name" on page 61.

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

security-name (SNMP Notifications)

Syntax security-name *security-name*;

Hierarchy Level [edit snmp v3 target-parameters *target-parameters-name* parameters]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the security name used when generating SNMP notifications.

Options *security-name*—If the USM security model is used, the security name identifies the user that is used when generating the notification. If the v1 or v2c security models are used, the security name identifies the SNMP community used when generating the notification.

NOTE: The access privileges for the group associated with this security name must allow this notification to be sent.

If you are using the v1 or v2 security models, the security name at the [edit snmp v3 vacm security-to-group] hierarchy level must match the security name at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

Usage Guidelines See "Configuring the Security Name" on page 70.

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

security-to-group

Syntax

```
security-to-group {
    security-model (usm | v1 | v2c) {
        security-name security-name;
            group group-name;
        }
    }
```

Hierarchy Level [edit snmp v3 vacm]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the group to which a specific security name belongs.

The remaining statements are explained separately.

Usage Guidelines See "Assigning Security Names to Groups" on page 60.

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

snmp-community

Syntax

```
snmp-community community-index {
    community-name community-name;
    security-name security-name;
    tag tag-name;
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the SNMP community.

Options *community-index*—(Optional) String that identifies an SNMP community.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the SNMP Community" on page 77.

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

tag

Syntax tag *tag-name*;

Hierarchy Level [edit snmp v3 notify *name*],
[edit snmp v3 snmp-community *community-index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a set of targets to receive traps or informs (for IPv4 packets only).

Options *tag-name*—Identifies the address of managers that are allowed to use a community string.

Usage Guidelines See "Configuring the Tag" on page 78 and "Configuring the Trap Notification" on page 63.

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

tag-list

Syntax tag-list [*tag-list*];

Hierarchy Level [edit snmp v3 target-address *target-address-names*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure an SNMP tag list used to select target addresses.

Options *tag-list*—Defines sets of target addresses.

Usage Guidelines See "Configuring the Tag List" on page 66.

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

target-address

Syntax

```
target-address target-address-name {
    address address;
    address-mask address-mask;
    inform-retry-count number;
    inform-timeout seconds;
    port <port-number>;
    tag-list [ tag-list ];
    target-parameters target-parameters-name;
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a management application's address and parameters to be used in sending notifications.

Options *target-address-name*—A string that identifies the target address.

The remaining statements are explained separately.

NOTE: You must configure the address mask when you configure the SNMP community.

Usage Guidelines See “Configuring the Trap Target Address” on page 65.

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

target-parameters

Syntax
```
target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
        message-processing-model (v1 | v2c | V3);
        security-model ( usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

}Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the message processing and security parameters to be used in sending notifications to a particular management target.

The remaining statements are explained separately.

Usage Guidelines See "Defining the Trap Target Parameters" on page 68.

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

type

Syntax type (trap | inform);

Hierarchy Level [edit snmp v3 notify *name*]

Release Information Statement introduced before JUNOS Release 7.4.
inform option added in JUNOS Release 7.4.

Description Configure the type of notification.

Options trap—Defines the type of notification as a trap. SNMP traps are unconfirmed notifications.

inform—Defines the type of notification as an inform. SNMP informs are confirmed notifications.

Usage Guidelines See "Configuring the Trap Notification" on page 63 and "Configuring SNMP Informs" on page 72.

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

user

Syntax user *username*;

Hierarchy Level [edit snmp v3 usm local-engine]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify a user associated with an SNMPv3 group.

Options *username*—SNMPv3 USM username.

Usage Guidelines See "Creating SNMPv3 Users" on page 49.

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

Usage Guidelines See "Creating SNMPv3 Users" on page 49.

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

usm

Syntax

```
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 {
          privacy-password privacy-password;
        privacy-none;
      }
    }
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure user-based security model (USM) information.

The remaining statements are explained separately.

Usage Guidelines See "Creating SNMPv3 Users" on page 49.

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

vacm

Syntax

```
vacm {
    access {
      group group-name {
        default-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;
        }
      }
    }
}
```

Hierarchy Level [edit snmp v3]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure view-based access control model (VACM) information.

The remaining statements are explained separately.

Usage Guidelines See "Defining Access Privileges for an SNMP Group" on page 55.

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

view

Syntax
```
view view-name {
    oid object-identifier (include | exclude);
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define a MIB view. A MIB view identifies a group of MIB objects. Each MIB object in a view has a common OID prefix. Each object identifier represents a subtree of the MIB object hierarchy. The view statement uses a view to specify a group of MIB objects on which to define access. To enable a view, you must associate the view with a community by including the view statement at the [edit snmp community *community-name*] hierarchy level. For SNMPv3, you must associate the view with a group name configured at the [edit snmp v3 vacm] hierarchy level.

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

Options *view-name*—Name of the view

The remaining statements are explained separately.

Usage Guidelines See “Configuring MIB Views” on page 54.

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

See Also “Associating MIB Views with an SNMP User Group” on page 58.

v3

Syntax

```
v3 {
    notify name {
      tag tag-name;
      type trap;
    }
    notify-filter profile-name {
      oid object-identifier (include | exclude);
    }
    snmp-community community-index {
      security-name community-name;
      security-name security-name;
      tag tag-name;
    }
    target-address target-address-name {
      address address;
      address-mask address-mask;
      inform-retry-count number;
      inform-timeout seconds;
      port <port-number>;
      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-model ( usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
      }
    }
    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-des {
            privacy-password privacy-password;
          }
          privacy-none;
        }
      }
```

```
vacm {
  access {
    group group-name {
      default-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;
      }
    }
  }
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure SNMPv3.

The remaining statements are explained separately.

Usage Guidelines See "Configuring SNMPv3" on page 45.

Required Privilege Level snmp—To view this statement in the configuration.

write-view

Syntax write-view *view-name*;

Hierarchy Level [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Associate the view with a community or a group name (SNMPv3).

Options *view-name*—The name of the view to which the SNMP user group has access.

Usage Guidelines See "Configuring the Write View" on page 59.

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

See Also See "Configuring MIB Views" on page 54.

Part 4

RMON Alarms and Events

Chapter 14

Configuring RMON Alarms and Events

The JUNOS software supports monitoring routers from remote devices. These values are measured against thresholds and trigger events when the thresholds are crossed. You configure remote monitoring (RMON) alarm and event entries to monitor the value of a Management Information Base (MIB) object.

For more information on configuring RMON alarm and event entries, see "Configuring RMON Alarms and Events" on page 185 and "Summary of RMON Alarm and Event Configuration Statements" on page 203.

For more information on monitoring integer-valued MIB objects, see "Monitoring RMON Alarms and Events" on page 193.

To configure RMON alarm and event entries, you include statements at the [edit snmp] hierarchy level of the configuration:

```
[edit snmp]
rmon {
    alarm index {
        description text-description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        rising-threshold integer;
        sample-type (absolute-value | delta-value);
        startup-alarm (falling-alarm | rising alarm | rising-or-falling-alarm);
        variable oid-variable;
    event index {
        community community-name;
        description description;
        type type;
    }
}
}
```

This chapter describes the minimum required configuration and discusses the following tasks for configuring RMON:

Minimum RMON Alarm and Event Entry Configuration

To enable RMON on the router, you must configure an alarm entry and an event entry. To do this, include the following statements at the [edit snmp rmon] hierarchy level:

```
[edit snmp rmon]
alarm index {
    rising-event-index index;
    rising-threshold integer;
    sample-type type;
    variable oid-variable;
}
event index;
```

Configuring an Alarm Entry and Its Attributes

An alarm entry monitors the value of a MIB variable. You can configure how often the value is sampled, the type of sampling to perform, and what event to trigger if a threshold is crossed.

This section discusses the following topics:

Configuring the Alarm Entry

An alarm entry monitors the value of a MIB variable. The rising-event-index, rising-threshold, sample-type, and variable statements are mandatory. All other statements are optional.

To configure the alarm entry, include the alarm statement and specify an index at the [edit snmp rmon] hierarchy level:

```
[edit snmp rmon]
alarm index {
    description description;
    falling-event-index index;
    falling-threshold integer;
    interval seconds;
    rising-event-index index;
    rising-threshold integer;
    sample-type (absolute-value | delta-value);
    startup-alarm (falling-alarm | rising alarm | rising-or-falling-alarm);
    variable oid-variable;
}
```

index is an integer that identifies an alarm or event entry.

Configuring the Description

The description is a text string that identifies the alarm entry.

To configure the description, include the description statement and a description of the alarm entry at the [edit snmp rmon alarm index] hierarchy level:

```
[edit snmp rmon alarm index]
description description;
```

Configuring the Falling Event Index or Rising Event Index

The falling event index identifies the event entry that is triggered when a falling threshold is crossed. The rising event index identifies the event entry that is triggered when a rising threshold is crossed.

To configure the falling event index or rising event index, include the falling-event-index or rising-event-index statement and specify an index at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
falling-event-index index;
rising-event-index index;
```

index can be from 0 through 65,535. The default for both the falling and rising event index is 0.

Configuring the Falling Threshold or Rising Threshold

The falling threshold is the lower threshold for the monitored variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup alarm is equal to falling-alarm or rising-or-falling-alarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising threshold. You must specify the falling threshold as an integer. Its default is 20 percent less than the rising threshold.

By default, the rising threshold is 0. The rising threshold is the upper threshold for the monitored variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated startup-alarm is equal to rising-alarm or rising-or-falling-alarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling threshold. You must specify the rising threshold as an integer.

NOTE: Before you configure the falling and rising threshold, check the file system on your PC card to determine its maximum file size. Use the value of the maximum size to determine how to configure the rising and falling thresholds.

To configure the falling threshold or rising threshold, include the falling-threshold or rising-threshold statement at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
falling-threshold integer;
rising-threshold integer;
```

integer can be a value from -2,147,483,647 through 2,147,483,647.

Configuring the Interval

The interval represents the period of time, in seconds, over which the monitored variable is sampled and compared with the rising and falling thresholds.

To configure the interval, include the interval statement and specify the number of seconds at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
interval seconds;
```

seconds can be a value from 1 through 2,147,483,647. The default is 60 seconds.

Configuring the Sample Type

The sample type identifies the method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absolute-value, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is delta-value, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds.

To configure the sample type, include the sample-type statement and specify the type of sample at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
sample-type (absolute-value | delta-value);
```

- absolute-value—Actual value of the selected variable is compared against the thresholds.
- delta-value—Difference between samples of the selected variable is compared against the thresholds.

Configuring the Startup Alarm

The startup alarm identifies the type of alarm that can be sent when this entry is first activated. You can specify it as falling-alarm, rising-alarm, or rising-or-falling-alarm.

To configure the startup alarm, include the startup-alarm statement and specify the type of alarm at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);
```

- falling-alarm—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.
- rising-alarm—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.
- rising-or-falling-alarm—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.

The default is rising-or-falling-alarm.

Configuring the Variable

The variable identifies the MIB object that is being monitored.

To configure the variable, include the variable statement and specify the object identifier or object name at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
variable oid-variable;
```

oid-variable is a dotted decimal (for example, 1.3.6.1.2.1.2.1.2.2.1.10.1) or MIB object name (for example, ifInOctets.1).

Configuring an Event Entry and Its Attributes

An event entry generates a notification for an alarm entry when its rising or falling threshold is crossed. You can configure the type of notification that is generated. To configure the event entry, include the event statement at the [edit snmp rmon] hierarchy level. All statements except the event statement are optional.

```
[edit snmp rmon]
event index {
    community community-name;
    description description;
    type type;
}
```

index identifies an entry event.

community-name is the trap group that is used when generating a trap. If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group. If nothing is configured, all the trap groups are examined, and traps are sent using each group with the rmon-alarm category set.

description is a text string that identifies the entry.

The *type* variable of an event entry specifies where the event is to be logged. You can specify the type as one of the following:

- log—Adds the event entry to the logTable.
- log-and-trap—Sends an SNMP trap and creates a log entry.
- none—Sends no notification.
- snmptrap—Sends an SNMP trap.

The default for the event entry type is log-and-trap.

Example: Configuring an RMON Alarm and Event Entry

Configure an RMON alarm and event entry:

```
[edit snmp]
    rmon {
      alarm 100 {
        description “input traffic on fxp0”;
        falling-event-index 100;
        falling-threshold 10000;
        interval 60;
        rising-event-index 100;
        rising-threshold 100000;
        sample-type delta-value;
        startup-alarm rising-or-falling-alarm;
        variable ifInOctets.1;
    }
      event 100 {
        community bedrock;
        description “emergency events”;
        type log-and-trap;
    }
}
```

Chapter 15

Monitoring RMON Alarms and Events

Use the remote monitoring (RMON) alarms and events feature to monitor integer-valued MIB objects, standard or enterprise-specific, on a Juniper Networks router. Configuration and operational information are in the MIB objects defined in alarmTable, eventTable, and logTable in RFC 2819. Additional information is defined by the Juniper Networks enterprise-specific extension to alarmTable defined in jnxRmonMIB (jnx-rmon-mib.txt).

This chapter covers the following main topics:

RMON Alarms

An RMON alarm identifies:

- A specific MIB object that is monitored.
- The frequency at which it is sampled.
- The method of sampling.
- The thresholds against which the monitored values are compared.

An RMON alarm can also identify a specific eventTable entry to be triggered when a threshold is crossed.

Configuration and operational values are defined in alarmTable in RFC 2819. Additional operational values are defined in Juniper Networks enterprise-specific extensions to alarmTable (jnxRmonAlarmTable).

This section covers the following topics:

alarmTable

alarmTable in the RMON MIB allows you to monitor and poll the following:

- alarmIndex—The index value for alarmTable that identifies a specific entry.
- alarmInterval—The interval, in seconds, over which data is sampled and compared with the rising and falling thresholds.
- alarmVariable—The MIB variable that is monitored by the alarm entry.
- alarmSampleType—The method of sampling the selected variable and calculating the value to be compared against the thresholds.
- alarmValue—The value of the variable during the last sampling period. This value is compared with the rising and falling thresholds.
- alarmStartupAlarm—The alarm sent when the entry is first activated.
- alarmRisingThreshold—The upper threshold for the sampled variable.
- alarmFallingThreshold—The lower threshold for the sampled variable.
- alarmRisingEventIndex—The eventTable entry used when a rising threshold is crossed.
- alarmFallingEventIndex—The eventTable entry used when a falling threshold is crossed.
- alarmStatus—Add and remove entries from the table. It can also be used to change the state of an entry to allow modifications.

NOTE: If this object is not set to valid, no action will be taken by the associated event alarm.

jnxRmonAlarmTable

The jnxRmonAlarmTable is a Juniper Networks enterprise-specific extension to alarmTable. It provides additional operational information and includes the following objects:

- jnxRmonAlarmGetFailCnt—The number of times the internal Get request for the variable monitored by this entry has failed.
- jnxRmonAlarmGetFailTime—The value of sysUpTime when an internal Get request for the variable monitored by this entry last failed.
- jnxRmonAlarmGetFailReason—The reason an internal Get request for the variable monitored by this entry last failed.
- jnxRmonAlarmGetOkTime—The value of sysUpTime when an internal Get request for the variable monitored by this entry succeeded and the entry left the getFailure state.
- jnxRmonAlarmState—The current state of this RMON alarm entry.

To view the Juniper Networks enterprise-specific extensions to the RMON alarm and event MIB, see www.juniper.net/techpubs/software/junos74/swconfig74-net-mgmt/html/mib-jnx-rmon.txt. For more information on the Juniper Networks enterprise-specific extensions to the RMON events and alarms MIB, see "Interpreting the Enterprise-Specific RMON Events and Alarms MIB" on page 343.

Using alarmTable to Monitor MIB Objects

To use alarmTable to monitor a MIB object, perform the following tasks:

Creating an Alarm Entry

To create an alarm entry, first create a new row in alarmTable using the alarmStatus object. For example, create alarm #1 using the UCD command-line utilities:

```
snmpset -Os -v2c router community alarmStatus.1 i createRequest
```

Configuring the Alarm MIB Objects

Once you have created the new row in alarmTable, configure the following alarm MIB objects:

NOTE: Other than alarmStatus, you cannot modify any of the objects in the entry if the associated alarmStatus object is set to valid.

alarmInterval

The interval, in seconds, over which data is sampled and compared with the rising and falling thresholds. For example, to set alarmInterval for alarm #1 to 30 seconds, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmInterval.1 i 30
```

alarmVariable

The object identifier of the variable to be sampled. During a Set request, if the supplied variable name is not available in the selected MIB view, a badValue error is returned. If at any time the variable name of an established alarmEntry is no longer available in the selected MIB view, the probe changes the status of alarmVariable to invalid. For example, to identify ifInOctets.61 as the variable to be monitored, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmVariable.1 o .1.3.6.1.2.1.2.2.1.10.61
```

alarmSampleType

The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absoluteValue, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is deltaValue, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds. For example, to set alarmSampleType for alarm #1 to deltaValue, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmSampleType.1 i deltaValue
```

alarmValue

The value of the variable during the last sampling period. This value is compared with the rising and falling thresholds. If the sample type is deltaValue, this value equals the difference between the samples at the beginning and end of the period. If the sample type is absoluteValue, this value equals the sampled value at the end of the period.

alarmStartupAlarm

An alarm that is sent when this entry is first set to valid. If the first sample after this entry becomes valid is greater than or equal to risingThreshold, and alarmStartupAlarm is equal to risingAlarm or risingOrFallingAlarm, then a single rising alarm is generated. If the first sample after this entry becomes valid is less than or equal to fallingThreshold and alarmStartupAlarm is equal to fallingAlarm or risingOrFallingAlarm, then a single falling alarm is generated. For example, to set alarmStartupAlarm for alarm #1 to risingOrFallingAlarm, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmStartupAlarm.1 i risingOrFallingAlarm
```

alarmRisingThreshold

A threshold for the sampled variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated alarmStartupAlarm is equal to risingAlarm or risingOrFallingAlarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches alarmFallingThreshold. For example, to set alarmRisingThreshold for alarm #1 to 100000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingThreshold.1 i 100000
```

alarmFallingThreshold

A threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated alarmStartupAlarm is equal to fallingAlarm or risingOrFallingAlarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches alarmRisingThreshold. For example, to set alarmFallingThreshold for alarm #1 to 10000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 10000
```

alarmOwner

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

alarmRisingEventIndex

The index of the eventEntry object that is used when a rising threshold is crossed. If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmRisingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingEventIndex.1 i 10
```

alarmFallingEventIndex

The index of the eventEntry object that is used when a falling threshold is crossed. If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmFallingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingEventIndex.1 i 10
```

Activating a New Row in alarmTable

To activate a new row in alarmTable, set alarmStatus to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

Modifying an Active Row in alarmTable

To modify an active row, first set alarmStatus to underCreation using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i underCreation
```

Then change the row contents using an SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 1000
```

Finally, activate the row by setting alarmStatus to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

Deactivating a Row in alarmTable

To deactivate a row in alarmTable, set alarmStatus to invalid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i invalid
```

RMON Events

An RMON event allows you to log the crossing of thresholds of other MIB objects. It is defined in eventTable for the RMON MIB.

This section covers the following topics:

eventTable

eventTable contains the following objects:

- eventIndex—An index that uniquely identifies an entry in eventTable. Each entry defines one event that will be generated when the appropriate conditions occur.
- eventDescription—A comment describing the event entry.
- eventType—Type of notification that the probe makes about this event.
- eventCommunity—Trap group used if an SNMP trap is to be sent. If eventCommunity is not configured, a trap is sent to each trap group configured with the rmon-alarm category.
- eventLastTimeSent—Value of sysUpTime when this event entry last generated an event.
- eventOwner—Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.
- eventStatus—Status of this event entry.

NOTE: If this object is not set to valid, no action is taken by the associated event entry. When this object is set to valid, all previous log entries associated with this entry (if any) will be deleted.

Using eventTable to Log Alarms

To use eventTable to log alarms, perform the following tasks:

- Creating an Event Entry on page 199
- Configuring the MIB Objects on page 200
- Activating the New Row in eventTable on page 201
- Deactivating a Row in eventTable on page 201

Creating an Event Entry

The RMON eventTable controls the generation of notifications from the router. Notifications can be logs (entries to logTable and syslogs) or SNMP traps. Each event entry can be configured to generate any combination of these notifications (or no notification). When an event specifies that an SNMP trap is to be generated, the trap group that is used when sending the trap is specified by the value of the associated eventCommunity object. Consequently, the community in the trap message will match the value specified by eventCommunity. If nothing is configured for eventCommunity, a trap is sent using each trap group that has the rmon-alarm category configured.

Configuring the MIB Objects

Once you have created the new row in eventTable, set the following objects:

- eventType on page 200
- eventCommunity on page 200
- eventOwner on page 201
- eventDescription on page 201

The eventType object is required. All other objects are optional.

eventType

The type of notification that the router generates when the event is triggered.

This object can be set to the following values:

- log—Adds the event entry to logTable.
- log-and-trap—Sends an SNMP trap and creates a log entry.
- none—Sends no notification.
- snmptrap—Sends an SNMP trap.

For example, to set eventType for event #1 to log-and-trap, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventType.1 i log-and-trap
```

eventCommunity

The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set. For example, to set eventCommunity for event #1 to boy-elroy, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventCommunity.1 s "boy-elroy"
```

NOTE: The eventCommunity object is optional. If you do not set this object, then the field is left blank.

eventOwner

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

For example, to set eventOwner for event #1 to george jetson, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventOwner.1 s "george jetson"
```

NOTE: The eventOwner object is optional. If you do not set this object, then the field is left blank.

eventDescription

Any text string specified by the creating management application or the CLI. The use of this string is application dependent.

For example, to set eventDescription for event #1 to spacelys sprockets, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventDescription.1 s "spacelys sprockets"
```

NOTE: The eventDescription object is optional. If you do not set this object, then the field is left blank.

Activating the New Row in eventTable

To activate the new row in eventTable, set eventStatus to valid using an SNMP Set request such as:

```
snmpset -Os -v2c router community eventStatus.1 i valid
```

Deactivating a Row in eventTable

To deactivate a row in eventTable, set eventStatus to invalid using an SNMP Set request such as:

```
snmpset -Os -v2c router community eventStatus.1 i invalid
```

Chapter 16

Summary of RMON Alarm and Event Configuration Statements

The following sections explain each of the remote monitoring (RMON) alarm and event configuration statements. The statements are organized alphabetically.

alarm

Syntax

```
alarm index {
    description description;
    falling-event-index index;
    falling-threshold integer;
    interval seconds;
    rising-event-index index;
    rising-threshold integer;
    sample-type (absolute-value | delta-value);
    startup-alarm (falling-alarm | rising-alarm | rising-or-falling alarm);
    variable oid-variable;
}
```

Hierarchy Level [edit snmp rmon]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure RMON alarm entries.

Options *index*—Identifies this alarm entry as an integer.

The remaining statements are explained separately.

Usage Guidelines See "Configuring an Alarm Entry and Its Attributes" on page 186.

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

See Also event on page 205

community

Syntax community *community-name*;

Hierarchy Level [edit snmp rmon event *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set.

Options *community-name*—Identifies the trap group that is used when generating a trap if the event is configured to send traps.

Usage Guidelines See "Configuring an Event Entry and Its Attributes" on page 190.

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

description

Syntax description *description*;

Hierarchy Level [edit snmp rmon alarm *index*],
[edit snmp rmon event *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Text description of alarm or event.

Options *description*—Text description of an alarm or event entry. If the description includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See "Configuring the Description" on page 187 and "Configuring an Event Entry and Its Attributes" on page 190.

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

event

Syntax
```
event index {
    community community-name;
    description description;
    type type;
}
```

Hierarchy Level [edit snmp rmon]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure RMON event entries.

Options *index*—Identifier for a specific event entry.

The remaining statements are explained separately.

Usage Guidelines See "Configuring an Event Entry and Its Attributes" on page 190.

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

See Also alarm on page 203

falling-event-index

Syntax falling-event-index *index*;

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The index of the event entry that is used when a falling threshold is crossed. If this value is zero, no event is triggered.

Options *index*—Index of the event entry that is used when a falling threshold is crossed.
Range: 0 through 65,535
Default: 0 seconds

Usage Guidelines See "Configuring the Falling Event Index or Rising Event Index" on page 187.

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

See Also rising-event-index on page 207

falling-threshold

Syntax falling-threshold *integer*;

Hierarchy Level [edit snmp rmon alarm *integer*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The lower threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup-alarm is equal to falling-alarm or rising-or-falling-alarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising-threshold.

Options *integer*—The lower threshold for the alarm entry.
Range: -2,147,483,648 through 2,147,483,647
Default: 20 percent less than rising-threshold

Usage Guidelines See "Configuring the Falling Threshold or Rising Threshold" on page 188.

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

See Also rising-threshold on page 207.

interval

Syntax interval *seconds*;

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Interval between samples.

Options *interval*—Time between samples, in seconds.
Range: 1 through 2,147,483,647 seconds
Default: 60 seconds

Usage Guidelines See "Configuring the Interval" on page 188.

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

rising-event-index

Syntax rising-event-index *index*;

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The index of the event entry that is used when a rising threshold is crossed. If this value is zero, no event is triggered.

Options *index*—Index of the event entry that is used when a rising threshold is crossed.
Range: 0 through 65,535
Default: 0

Usage Guidelines See "Configuring the Falling Event Index or Rising Event Index" on page 187.

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

See Also falling-event-index on page 205

rising-threshold

Syntax rising-threshold *integer*;

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The upper threshold for the sampled variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated startup-alarm is equal to falling-alarm or rising-or-falling-alarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling-threshold.

Options *integer*—The lower threshold for the alarm entry.
Range: -2,147,483,648 through 2,147,483,647
Default: 0

Usage Guidelines See "Configuring the Falling Threshold or Rising Threshold" on page 188.

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

See Also falling-threshold on page 206

rmon

Syntax rmon { ... }

Hierarchy Level [edit snmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure Remote Monitoring.

Usage Guidelines See "Configuring RMON Alarms and Events" on page 185.

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

sample-type

Syntax sample-type (absolute-value | delta-value);

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Method of sampling the selected variable.

Options
- absolute-value—Actual value of the selected variable is used when comparing against the thresholds.
- delta-value—Difference between samples of the selected variable is used when comparing against the thresholds.

Usage Guidelines See "Configuring the Sample Type" on page 189.

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

startup-alarm

Syntax startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The alarm that can be sent upon entry startup.

Options falling-alarm—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.

rising-alarm—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.

rising-or-falling-alarm—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.

Default: rising-or-falling-alarm

Usage Guidelines See "Configuring the Startup Alarm" on page 189.

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

type

Syntax type *type*;

Hierarchy Level [edit snmp rmon event *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Type of notification generated when a threshold is crossed.

Options

- *type*—Type of notification. It can be one of the following:
  - log—Add an entry to logTable.
  - log-and-trap—Send an SNMP trap and make a log entry.
  - none—No notifications are sent.
  - snmptrap—Send an SNMP trap.

Default: log-and-trap

Usage Guidelines See "Configuring an Event Entry and Its Attributes" on page 190.

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

variable

Syntax variable *oid-variable*;

Hierarchy Level [edit snmp rmon alarm *index*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Object identifier (OID) of MIB variable to be monitored.

Options *oid-variable*—OID of the MIB variable that is being monitored. The OID can be a dotted decimal (for example, 1.3.6.1.2.1.2.1.2.2.1.10.1) or use the MIB objects name (for example, ifInOctets.1).

Usage Guidelines See "Configuring the Variable" on page 189.

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

Part 5
Monitoring Service Quality

Chapter 17

Monitoring Service Quality in Service Provider Networks

This chapter provides guidelines for monitoring the service quality of an IP network. It describes how service providers and network administrators can use information provided by Juniper Networks routers to monitor network performance and capacity. This chapter assumes you have a thorough understanding of the Simple Network Management Protocol (SNMP) and the associated Management Information Base (MIB) supported by the JUNOS software.

NOTE: For a good introduction to the process of monitoring an IP network, see RFC 2330, *Framework for IP Performance Metrics.*

This chapter includes the following topics:

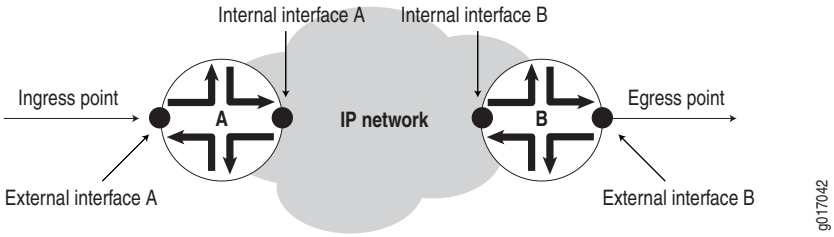
Measurement Points

Defining the measurement points where metrics are measured is equally as important as defining the metrics themselves. This section describes measurement points within the context of this chapter and helps identify where measurements can be taken from a service provider network. It is important to understand exactly where a measurement point is. Measurement points are vital to understanding the implication of what the actual measurement means.

An IP network consists of a collection of routers connected via physical links that are all running the Internet Protocol. As an entity, you can view the network as a collection of routers with an ingress (entry) point and an egress (exit) point. See Figure 2.

- Network-centric measurements are taken at measurement points that most closely map to the ingress and egress points for the network itself. For example, to measure delay across the provider network from site A to site B, the measurement points should be the ingress point to the provider network at site A and the egress point at site B.
- Router-centric measurements are taken directly from the routers themselves, but be careful to insure that the correct router subcomponents have been identified in advance.

Figure 2: Network Entry Points



NOTE: Figure 2 does not show the client networks at customer premises, but they would be located on either side of the ingress and egress points. Although this chapter does not discuss how to measure network services as perceived by these client networks, you can use measurements taken for the service provider network as input into such calculations.

This section includes the following topics:

Basic Key Performance Indicators

For example, you could monitor a service provider network for three basic key performance indicators (KPIs):

- *Availability* measures the "reachability" of one measurement point from another measurement point at the network layer (for example, using ICMP ping). The underlying routing and transport infrastructure of the provider network will support the availability measurements, with failures highlighted as unavailability.
- *Health* measures the number and type of errors that are occurring on the provider network, and can consist of both router-centric and network-centric measurements, such as hardware failures or packet loss.
- *Performance* of the provider network measures how well it can support IP services (for example, in terms of delay or utilization).

Each KPI is defined in more detail later in this chapter.

Setting Baselines

How well is the provider network performing? We recommend an initial three-month period of monitoring to identify a network's normal operational parameters. With this information, you can recognize exceptions and identify abnormal behavior. You should continue baseline monitoring for the lifetime of each measured metric. Over time, you will be able to recognize performance trends and growth patterns.

Within the context of this chapter, many of the metrics identified do not have an allowable operational range associated with them. In most cases, you cannot identify the allowable operational range until you have determined a baseline for the actual variable on a specific network.

Remote Monitoring

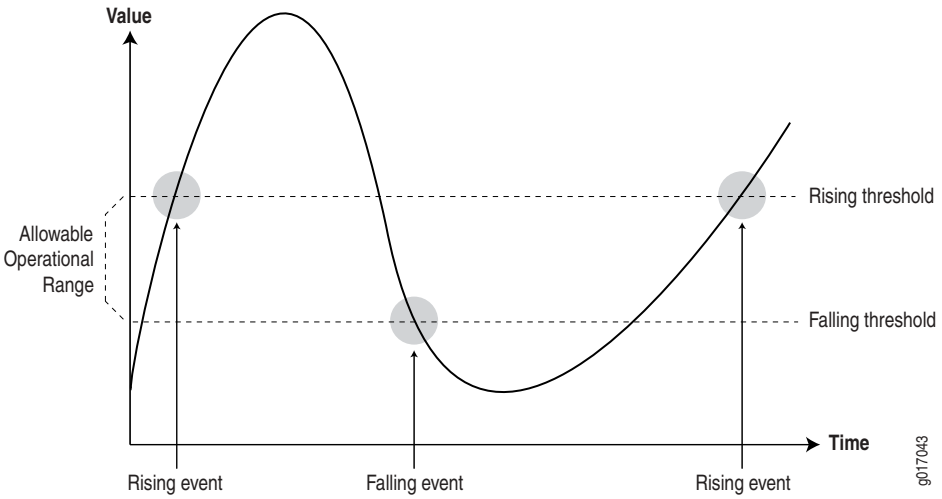
Health and performance monitoring can benefit from the remote monitoring of SNMP variables by the local SNMP agents running on each router. The SNMP agents compare MIB values against predefined thresholds and generate exception alarms without the need for polling by a central SNMP management platform. This is an effective mechanism for proactive management, as long as the thresholds have baselines determined and set correctly. For more information, see RFC 2819, *Remote Network Monitoring MIB*.

This section includes the following topics:

Setting Thresholds

By setting a rising and a falling threshold for a monitored variable, you can be alerted whenever the value of the variable falls outside of the allowable operational range. (See Figure 3.)

Figure 3: Setting Thresholds



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

To configure remote monitoring, specify the following pieces of information:

- The variable to be monitored (by its SNMP object identifier)
- The frequency (in time) between each inspection
- A rising threshold
- A falling threshold
- A rising event
- A falling event

Before you can successfully 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 three months is not unusual when first identifying the operational ranges and defining thresholds, but baseline monitoring should continue over the life span of each monitored variable.

RMON Command-Line Interface

The JUNOS software provides two mechanisms you use to control the Remote Monitoring agent on the router: command-line interface (CLI) and SNMP. To configure an RMON entry using the CLI, enter the following JUNOS commands at the [edit snmp] hierarchy level:

```
rmon {
    alarm index {
        description description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        rising-threshold integer;
        sample-type (absolute-value | delta-value);
        startup-alarm (falling | rising | rising-or-falling);
        variable oid-variable;
        }
    event index {
        community community-name;
        description description;
        type (log | trap | log-and-trap | none);
    }
}
```

If you do not have CLI access, you can configure remote monitoring using the SNMP Manager or management application, assuming SNMP access has been granted. (See Table 14.) To configure RMON via SNMP, perform SNMP SETs to the RMON event and alarm tables.

RMON Event Table

Set up an event for each type that you want to generate. For example, you could have two generic events, rising and falling, or many different events for each variable that is being monitored (for example, temperature rising event, temperature falling event, Firewall hit event, Interface utilization event, and so on). Once configured, you do not need to update these events.

Table 14: RMON Event Table

Field	Description
eventDescription	A textual description of this event.
eventType	Type of event (for example, log, trap, or log and trap).
eventCommunity	The trap-group to which to send this event (as defined in the JUNOS software configuration, which is not the same as the community).
eventOwner	The entity (for example, manager) that created this event.
eventStatus	Status of this row (for example, valid, invalid, or createRequest).

RMON Alarm Table

The RMON alarm table stores the SNMP object identifiers (including their instances) of the variables that are being monitored, together with any rising and falling thresholds and their corresponding event indexes. To create an RMON request, specify the fields shown in Table 15.

Table 15: RMON Alarm Table

Field	Description
alarmStatus	Status of this row (for example, valid, invalid, or createRequest)
alarmInterval	Sampling period (in seconds) of the monitored variable
alarmVariable	The OID (and instance) of the variable to be monitored
alarmValue	The actual value of the sampled variable
alarmSampleType	Look at either absolute or delta changes
alarmStartupAlarm	Initial alarm (rising, falling, or either)
alarmRisingThreshold	Rising threshold against which to compare the value
alarmFallingThreshold	Falling threshold against which to compare the value
alarmRisingEventIndex	Index (row) of the rising event in the event table
alarmFallingEventIndex	Index (row) of the falling event in the event table

Both the alarmStatus and eventStatus fields are entryStatus primitives, as defined in RFC 2579, *Textual Conventions for SMIv2*.

Troubleshooting RMON

You troubleshoot the RMON agent, rmopd, that runs on the router by inspecting the contents of the Juniper Networks enterprise RMON MIB, jnxRmon, which provides the extensions listed in Table 16 to the RFC 2819 alarmTable.

Table 16: jnxRmon Alarm Extensions

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

Monitoring the extensions in this table provides clues as to why remote alarms may be not behave as expected.

Configuring SNMP

This section shows the basic JUNOS configuration required to configure SNMP version 2 on each router.

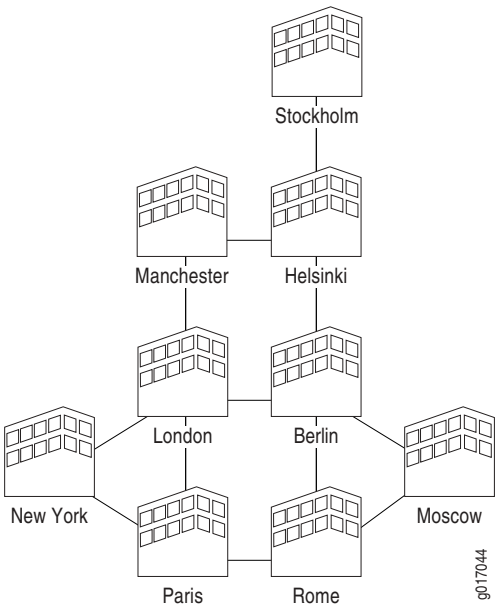
```
[edit]
snmp {
    community community-name {
        authorization authorization;
        view view-name;
    }

    trap-group group-name {
        targets {
            address;
        }
        version v2;

    view view-name {
        oid .<object-identifier> include;
    }
}
```

For more information, see “Configuring SNMP” on page 25.

Definition of Network Availability

Availability of a service provider's IP network can be thought of as the "reachability" between the regional points of presence (POP), as shown in Figure 4.

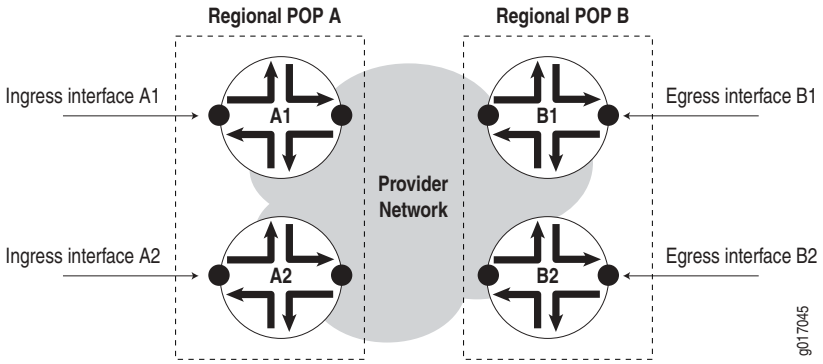
Figure 4: Regional Points of Presence



With the example above, when you use a full mesh of measurement points, where every POP measures the availability to every other POP, you can calculate the total availability of the service provider's network. This KPI can also be used to help monitor the service level of the network, and can be used by the service provider and their customers to determine if they are operating within the terms of their service-level agreement (SLA).

Where a POP may consist of multiple routers, take measurements to each router as shown in Figure 5.

Figure 5: Measurements to Each Router



Measurements include:

- Path availability—Availability of an egress interface B1 as seen from an ingress interface A1.
- Router availability—Percentage path availability of all measured paths terminating on the router.
- POP availability—Percentage router availability between any two regional POPs, A and B.
- Network availability—Percentage POP availability for all regional POPs in the service provider's network.

To measure POP availability of POP A to POP B in Figure 5, you must measure the following four paths:

Path A1 => B1
Path A1 => B2
Path A2 => B1
Path A2 => B2

Measuring availability from POP B to POP A would require a further four measurements, and so on.

A full mesh of availability measurements can generate a lot of management traffic. From the example diagram above:

- Each POP has two co-located provider edge (PE) routers each with 2xSTM1 interfaces, for a total of 18 PE routers and 36xSTM1 interfaces.
- There are six core provider (P) routers, four with 2xSTM4 and 3xSTM1 interfaces each, and two with 3xSTM4 and 3xSTM1 interfaces each.

This makes a total of 68 interfaces. A full mesh of paths between every interface is:

$$\frac{n \times (n-1)}{2} \quad \text{gives} \quad \frac{68 \times (68-1)}{2} \quad = 2278 \ \text{paths}$$

To reduce management traffic on the service provider's network, instead of generating a full mesh of interface availability tests (for example, from each interface to every other interface), you can measure from each router's loopback address. This reduces the number of availability measurements required to a total of one for each router, or:

$$\frac{n \times (n-1)}{2} \quad \text{gives} \quad \frac{24 \times (24-1)}{2} \quad = 276 \ \text{measurements}$$

This measures availability from each router to every other router.

Monitoring the SLA and the Required Bandwidth

A typical SLA between a service provider and a customer might state:

> A Point of Presence is the connection of two back-to-back provider edge routers to separate core provider routers using different links for resilience. The system is considered to be unavailable when either an entire POP becomes unavailable or for the duration of a Priority 1 fault.

An SLA availability figure of 99.999 percent for a provider's network would relate to a down time of approximately 5 minutes per year. Therefore, to measure this proactively, you would have to take availability measurements at a granularity of less than one every five minutes. With a standard size of 64 bytes per ICMP ping request, one ping test per minute would generate 7680 bytes of traffic per hour per destination, including ping responses. A full mesh of ping tests to 276 destinations would generate 2,119,680 bytes per hour, which represents the following:

- On an OC3/STM1 link of 155.52 Mpbs, a utilization of 1.362 percent
- On an OC12/STM4 link of 622.08 Mpbs, a utilization of 0.340 percent

With a size of 1500 bytes per ICMP ping request, one ping test per minute would generate 180,000 bytes per hour per destination, including ping responses. A full mesh of ping tests to 276 destinations would generate 49,680,000 bytes per hour, which represents the following:

- On an OC3/STM1 link, 31.94 percent utilization
- On an OC12/STM4 link, 7.986 percent utilization

Each router can record the results for every destination tested. (See "Juniper Networks Proxy Ping" on page 223.) With one test per minute to each destination, a total of 1 x 60 x 24 x 276 = 397,440 tests per day would be performed and recorded by each router. All ping results are stored in the pingProbeHistoryTable (see RFC 2925) and can be retrieved by an SNMP performance reporting application (for example, service performance management software from InfoVista, Inc., or Concord Communications, Inc.) for post processing. This table has a maximum size of 4,294,967,295 rows, which is more than adequate.

Measuring Availability

There are two methods you can use to measure availability:

- Proactive—Availability is automatically measured as often as possible by an operational support system.
- Reactive—Availability is recorded by a Help desk when a fault is first reported by a user or a fault monitoring system.

This section discusses technical solutions for a proactive monitoring solution. Topics include:

- Juniper Networks Proxy Ping on page 223
- Cisco Systems Service Assurance Agent on page 225

Juniper Networks Proxy Ping

A simple indication of availability is to measure the reachability of a remote site by using an application such as ping. The Distributed Management MIB defined in RFC 2925 provides a mechanism to configure remote ping tests using SNMP. Using this mechanism, you can configure a router to remotely run regular ping tests to other hosts and record the results in its MIB.

Figure 6: Ping Tests to Other Hosts

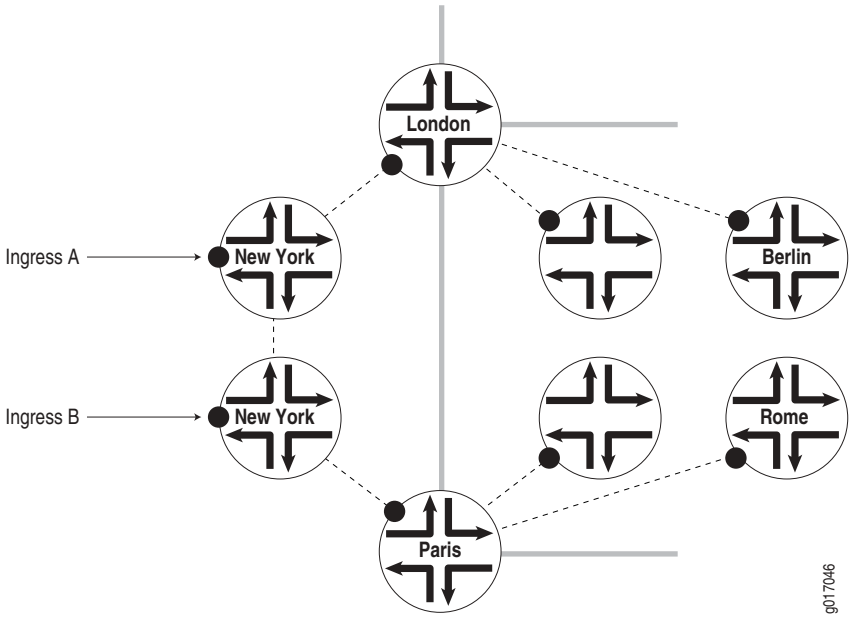


Figure 6 shows the two New York PE routers pinging from their ingress interfaces to remote routers at other POPs. You can use the loopback interface on each router as the destination address to test availability, instead of specifying individual interfaces.

The advantage of using this feature in JUNOS software is that you can run these ping tests on remote routers instead of from a single central management station. This means that the results indicate availability from the remote routers to their destinations, rather than from the management station to the destinations. In addition to distributing the monitoring in this way, a Juniper Networks enhancement to the Ping MIB allows ping tests to be injected into Layer 3 VPNs (as defined in RFC 2547), thereby allowing the measurements to become VPN aware.

Configuring Proxy Ping Measurement Tests

To configure remote ping requests, the SNMP manager creates entries in the pingCtlTable (.1.3.6.1.2.1.80.1) of each router. These fields are defined in the pingCtlTable table, shown in Table 17.

Table 17: Fields in pingCtlTable

Field	Description
pingCtlOwnerIndex	Variable-length owner name (compound index key)
pingCtlTestName	Variable-length test name (compound index key)
pingCtlSourceAddress	External interface A
pingCtlTargetAddress	External interface B
pingCtlTargetAddressType	IPv4 or IPv6
pingCtlDataSize	Size, in bytes, of each ping probe (64 KB max)
pingCtlProbeCount	Number of probes in this test
pingCtlAdminStatus	Enable or disable this test
pingCtlFrequency	Time delay between each probe
pingCtlTrapGeneration	Generate traps on probe or test failure
pingCtlType	ICMP or UDP echo
pingCtlDSField	DiffServ (ToS) byte setting
pingCtlRowStatus	Current status of this row

Additional Juniper Networks enterprise Ping MIB extensions to the control table allow the SNMP manager to specify the exact VRF or routing instance to use. This allows the pings to be injected into VPNs, thereby measuring the availability as seen within the VPN. The additional jnxPingCtlTable contains the fields shown in Table 18.

Table 18: Fields in jnxPingCtlTable

Field	Description
jnxPingCtlIfName	Outgoing interface to use
jnxPingCtlRoutingIfIndex	Outgoing logical interface ifIndex to use
jnxPingCtlRoutingIfName	Outgoing routing instance to use
jnxPingCtlRoutingInstanceName	VRF to use

The router creates entries by first setting the pingCtlRowStatus field to 5 (create) and populating the fields in the row with the required values. Changing the value of pingCtlRowStatus to 1 activates the entry, and setting the pingCtlAdminStatus to 1 enables the entry, thereby running it. The pingCtlRowStatus field is a RowStatus primitive, as defined in RFC 2579, *Textual Conventions for SMIv2*.

Proxy Ping Availability Results

For each test, the router creates a row in the pingResultsTable table, which contains the notable fields shown in Table 19.

Table 19: Fields in pingResultsTable

Field	Description
pingCtlOwnerIndex	Variable-length owner name (compound index key)
pingCtlTestName	Variable-length test name (compound index key)
pingResultsIpTargetAddress	IP address of remote destination
pingResultsMinRtt	Minimum round-trip time for this record, in milliseconds
pingResultsMaxRtt	Maximum round-trip time, in milliseconds
pingResultsAverageRtt	Mean round-trip time for this record, in milliseconds
pingResultsProbeResponses	Number of (ping) probe responses received
pingResultsSentProbes	Number of (ping) probes sent
pingResultsRttSumofSquares	Used to calculate standard deviation for this record
pingResultsLastGoodProbe	Date and time the last response was received

For each ping probe, a row is also created in the pingProbeHistoryTable table with the notable fields shown in Table 20.

Table 20: Fields in pingProbeHistoryTable

Field	Description
pingCtlOwnerIndex	Variable-length owner name (compound index key)
pingCtlTestName	Variable-length test name (compound index key)
pingProbeHistoryIndex	Unique index number in this table
pingProbeHistoryResponse	The time, in milliseconds, that this probe took
pingProbeHistoryTime	The date and time when this result was recorded

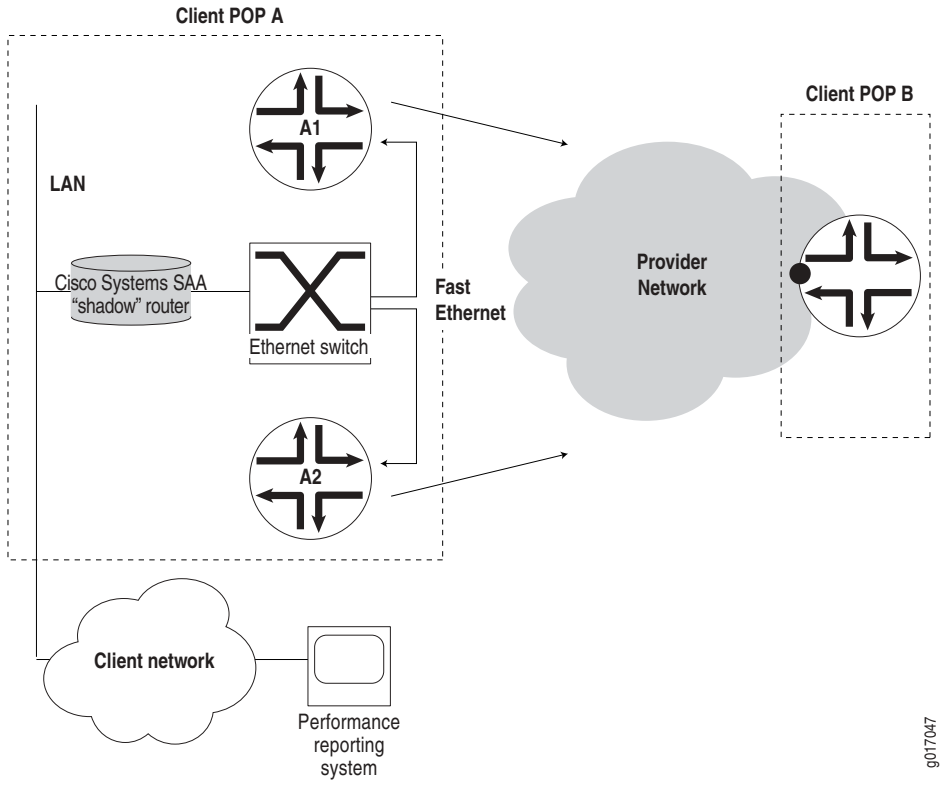
Cisco Systems Service Assurance Agent

An alternative mechanism adopted by some service provider customers is to use Cisco Systems Service Assurance Agent (SAA) from their own client networks. The advantages of using Cisco Systems SAA are:

- Customers can monitor their own service quality.
- Reporting applications from InfoVista, Inc., or Concord Communications, Inc., are already available.
- Other applications (to ICMP) can be measured.

A typical Cisco Systems SAA architecture used by service provider customers is shown in Figure 7.

Figure 7: Cisco Systems SAA Architecture



This architecture assumes the following:

- A Cisco Systems SAA router is available on the client network or POP.
- An Ethernet switch is available to connect the SAA router to the service provider network.
- The SAA router has sufficient Ethernet ports to connect to each Juniper Networks router.
- Each Juniper Networks router has a free WAN port on a Fast Ethernet PIC.
- The availability test is generated by the SAA agent running on the Cisco Systems router.
- The performance report system has access to the SAA routers, via the client's own network.

Measuring Health

You can monitor health metrics reactively by using fault management software such as SMARTS InCharge, Micromuse Netcool Omnibus, or Concord Live Exceptions. We recommend that you monitor the health metrics shown in Table 21.

Table 21: Health Metrics

Metric:	Errors In
Description	Number of inbound packets that contained errors, preventing them from being delivered.
MIB Name	IF-MIB (RFC 2233)
Variable Name	ifInErrors
Variable OID	.1.3.6.1.31.2.2.1.14
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	Logical interfaces

Metric:	Errors Out
Description	Number of outbound packets that contained errors, preventing them from being transmitted.
MIB Name	IF-MIB (RFC 2233)
Variable Name	ifOutErrors
Variable OID	.1.3.6.1.31.2.2.1.20
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	Logical interfaces

Metric:	Discards In
Description	Number of inbound packets discarded, even though no errors were detected.
MIB Name	IF-MIB (RFC 2233)
Variable Name	ifInDiscards
Variable OID	.1.3.6.1.31.2.2.1.13
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	Logical interfaces

Metric:	Unknown Protocols
Description	Number of inbound packets discarded because they were of an unknown protocol.
MIB Name	IF-MIB (RFC 2233)
Variable Name	ifInUnknownProtos
Variable OID	.1.3.6.1.31.2.2.1.15
Frequency (mins)	60

Table 21: Health Metrics (continued)

Allowable Range	To be baselined
Managed Objects	Logical interfaces
Metric:	Interface Operating Status
Description	Operational status of an interface.
MIB Name	IF-MIB (RFC 2233)
Variable Name	ifOperStatus
Variable OID	.1.3.6.1.31.2.2.1.8
Frequency (mins)	15
Allowable Range	1 (up)
Managed Objects	Logical interfaces
Metric:	Label Switched Path (LSP) State
Description	Operational state of an MPLS label-switched path.
MIB Name	MPLS-MIB
Variable Name	mplsLspState
Variable OID	mplsLspEntry.2
Frequency (mins)	60
Allowable Range	2 (up)
Managed Objects	All label-switched paths in the network.
Metric:	Component Operating Status
Description	Operational status of a router hardware component.
MIB Name	JUNIPER-MIB
Variable Name	jnxOperatingState
Variable OID	.1.3.6.1.4.1.2636.1.13.1.6
Frequency (mins)	60
Allowable Range	2 (running) or 3 (ready)
Managed Objects	All components in each Juniper Networks router
Metric:	Component Operating Temperature
Description	Operational temperature of a hardware component, in Celsius.
MIB Name	JUNIPER-MIB
Variable Name	jnxOperatingTemp
Variable OID	.1.3.6.1.4.1.2636.1.13.1.7
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All components in a chassis

Table 21: Health Metrics (continued)

Metric:	System Up Time
Description	Time, in milliseconds, that the system has been operational.
MIB Name	MIB-2 (RFC 1213)
Variable Name	sysUpTime
Variable OID	.1.3.6.1.1.3
Frequency (mins)	60
Allowable Range	Increasing only (decrement indicates a restart)
Managed Objects	All routers

Metric:	No IP Route Errors
Description	Number of packets that could not be delivered because there was no IP route to their destination.
MIB Name	MIB-2 (RFC 1213)
Variable Name	ipOutNoRoutes
Variable OID	ip.12
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	Each router

Metric:	Wrong SNMP Community Names
Description	Number of incorrect SNMP community names received.
MIB Name	MIB-2 (RFC 1213)
Variable Name	snmpInBadCommunityNames
Variable OID	snmp.4
Frequency (hours)	24
Allowable Range	To be baselined
Managed Objects	Each router

Metric:	SNMP Community Violations
Description	Number of valid SNMP communities used to attempt invalid operations (for example, attempting to perform SNMP SET requests).
MIB Name	MIB-2 (RFC 1213)
Variable Name	snmpInBadCommunityUses
Variable OID	snmp.5
Frequency (hours)	24
Allowable Range	To be baselined
Managed Objects	Each router

Table 21: Health Metrics (continued)

Metric:	Redundancy Switch Over
Description	Total number of redundancy switchovers reported by this entity.
MIB Name	JUNIPER-MIB
Variable Name	jnxRedundancySwitchoverCount
Variable OID	jnxRedundancyEntry.8
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All Juniper Networks routers with redundant Routing Engines

Metric:	FRU State
Description	Operational status of each field-replaceable unit (FRU).
MIB Name	JUNIPER-MIB
Variable Name	jnxFruState
Variable OID	jnxFruEntry.8
Frequency (mins)	15
Allowable Range	2 through to 6 for ready/online states. See jnxFruOfflineReason in the event of a FRU failure.
Managed Objects	All FRUs in all Juniper Networks routers.

Metric:	Rate of Tall Dropped Packets
Description	Rate of tail-dropped packets per output queue, per forwarding class, per interface.
MIB Name	JUNIPER-COS-MIB
Variable Name	jnxCosIfqTailDropPktRate
Variable OID	jnxCosIfqStatsEntry.12
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	For each forwarding class/interface in the provider network, when CoS is enabled.

SNMP traps are also a good mechanism to use for health management. For more information, see “Standard SNMP Traps” on page 119 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 111.

Measuring Performance

The performance of a service provider's network is usually defined as how well it can support services, and is measured with metrics such as delay and utilization. We suggest that you monitor the following performance metrics using applications such as InfoVista Service Performance Management or Concord Network Health (see Table 22).

Table 22: Performance Metrics

Metric:	Average Delay
Description	Average round-trip time (in milliseconds) between two measurement points.
MIB Name	DISMAN-PING-MIB (RFC 2925)
Variable Name	pingResultsAverageRtt
Variable OID	pingResultsEntry.6
Frequency (mins)	15 (or depending upon ping test frequency)
Allowable Range	To be baselined
Managed Objects	Each measured path in the network

Metric:	Interface Utilization
Description	Utilization percentage of a logical connection.
MIB Name	IF-MIB
Variable Name	(ifInOctets & ifOutOctets) * 8 / ifSpeed
Variable OID	ifTable entries
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All operational interfaces in the network

Metric:	Disk Utilization
Description	Utilization of disk space within the Juniper Networks router
MIB Name	HOST-RESOURCES-MIB (RFC 2790)
Variable Name	hrStorageSize – hrStorageUsed
Variable OID	hrStorageEntry.5 – hrStorageEntry.6
Frequency (mins)	1440
Allowable Range	To be baselined
Managed Objects	All Routing Engine hard disks

Table 22: Performance Metrics (continued)

Metric:	Memory Utilization
Description	Utilization of memory on the Routing Engine and FPC.
MIB Name	JUNIPER-MIB (Juniper enterprise Chassis MIB)
Variable Name	jnxOperatingHeap
Variable OID	Table for each component
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All Juniper Networks routers

Metric:	CPU Load
Description	Average utilization over the past minute of a CPU.
MIB Name	JUNIPER-MIB (Juniper enterprise Chassis MIB)
Variable Name	jnxOperatingCPU
Variable OID	Table for each component
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All Juniper Networks routers

Metric:	LSP Utilization
Description	Utilization of the MPLS label-switched path.
MIB Name	MPLS-MIB
Variable Name	mplsPathBandwidth / (mplsLspOctets * 8)
Variable OID	mplsLspEntry.21 and mplsLspEntry.3
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	All label-switched paths in the network

Metric:	Output Queue Size
Description	Size, in packets, of each output queue per forwarding class, per interface.
MIB Name	JUNIPER-COS-MIB
Variable Name	jnxCosIfqQedPkts
Variable OID	jnxCosIfqStatsEntry.3
Frequency (mins)	60
Allowable Range	To be baselined
Managed Objects	For each forwarding class/interface in the network, once CoS enabled.

This section includes the following topics:

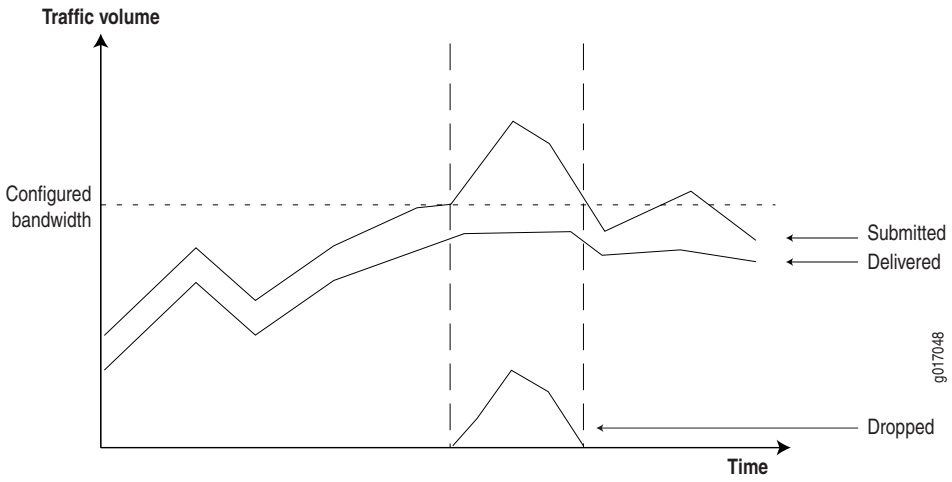
Measuring Class of Service

You can use class-of-service (CoS) mechanisms to regulate how certain classes of packets are handled within your network during times of peak congestion. Typically you must perform the following steps when implementing a class-of-service mechanism:

- Identify the type of packets that will be applied to this class. For example, include all customer traffic from a specific ingress edge interface within one class, or include all packets of a particular protocol such as voice over IP (VoIP).
- Identify the required deterministic behavior for each class. For example, if VoIP is important, give VoIP traffic the highest priority during times of network congestion. Conversely, you can downgrade the importance of Web traffic during congestion, as it may not impact customers too much.

With this information, you can configure mechanisms at the network ingress to monitor, mark, and police traffic classes. Marked traffic can then be handled in a more deterministic way at egress interfaces, typically by applying different queuing mechanisms for each class during times of network congestion. You can collect information from the network to provide customers with reports showing how the network is behaving during times of congestion. (See Figure 8.)

Figure 8: Network Behavior During Congestion



To generate these reports, routers must provide the following information:

- Submitted traffic—Amount of traffic received per class.
- Delivered traffic—Amount of traffic transmitted per class.
- Dropped traffic—Amount of traffic dropped because of CoS limits.

The following section outlines how this information is provided by Juniper Networks routers.

Inbound Firewall Filter Counters per Class

Firewall filter counters are a very flexible mechanism you can use to match and count inbound traffic per class, per interface. For example:

```
firewall {
    filter f1 {
        term t1 {
            from {
                dscp af11;
            }
            then {
                # Assured forwarding class 1 drop profile 1 count inbound-af11;
                accept;
            }
        }
    }
}
```

For example, Table 23 shows additional filters used to match the other classes.

Table 23: Inbound Traffic Per Class

DSCP Value	Firewall Match Condition	Description
10	af11	Assured forwarding class 1 drop profile 1
12	af12	Assured forwarding class 1 drop profile 2
18	af21	Best effort class 2 drop profile 1
20	af22	Best effort class 2 drop profile 2
26	af31	Best effort class 3 drop profile 1

Any packet with a CoS DiffServ code point (DSCP) conforming to RFC 2474 can be counted in this way. The Juniper Networks enterprise firewall filter MIB presents the counter information in the variables shown in Table 24.

Table 24: Inbound Counters

Indicator Name:	Inbound Counters
MIB	jnxFirewalls
Table	jnxFirewallCounterTable
Index	jnxFWFilter.jnxFWCounter
Variables	jnxFWCounterPacketCount jnxFWCounterByteCount
Description	Number of bytes being counted pertaining to the specified firewall filter counter.
SNMP Version	SNMPv2

This information can be collected by any SNMP management application that supports SNMPv2. Products from vendors such as Concord Communications, Inc., and InfoVista, Inc., provide support for the Juniper Networks Firewall MIB with their native Juniper Networks device drivers.

Monitoring Output Bytes per Queue

You can use the Juniper Networks enterprise ATM CoS MIB to monitor outbound traffic, per virtual circuit forwarding class, per interface. (See Table 25.)

Table 25: Outbound Counters for ATM Interfaces

Indicator Name:	Outbound Counters
MIB	JUNIPER-ATM-COS-MIB
Variable	jnxCosAtmVcQstatsOutBytes
Index	ifIndex.atmVclVpi.atmVclVci.jnxCosFcId
Description	Number of bytes belonging to the specified forwarding class that were transmitted on the specified virtual circuit.
SNMP Version	SNMPv2

Non-ATM interface counters are provided by the Juniper Networks enterprise CoS MIB, which provides information shown in Table 26.

Table 26: Outbound Counters for Non-ATM Interfaces

Indicator Name:	Outbound Counters
MIB	JUNIPER-COS-MIB
Table	jnxCosIfqStatsTable
Index	jnxCosIfqIfIndex.jnxCosIfqFc
Variables	jnxCosIfqTxedBytes jnxCosIfqTxedPkts
Description	Number of transmitted bytes/packets per interface per forwarding class.
SNMP Version	SNMPv2

Dropped Traffic

You can calculate the amount of dropped traffic by submitting the outbound traffic from the incoming traffic:

Dropped = Inbound Counter – Outbound Counter

You can also select counters from the CoS MIB, as shown in Table 27.

Table 27: Dropped Traffic Counters

Indicator Name:	Dropped Traffic
MIB	JUNIPER-COS-MIB
Table	jnxCosIfqStatsTable
Index	jnxCosIfqIfIndex.jnxCosIfqFc
Variables	jnxCosIfqTailDropPkts jnxCosIfqTotalRedDropPkts
Description	The number of tail dropped or RED dropped packets per interface per forwarding class
SNMP Version	SNMPv2

Part 6

Juniper Networks Enterprise-Specific MIBs

Chapter 18

Interpreting the Structure of Management Information MIB

The structure of management information MIB defines the top-level structure of the Juniper Networks enterprise-specific MIB space. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-smi.txt.

The structure of management information MIB space has five root branches:

jnxProducts

The object identifier for the jnxProducts root branch of the structure of management information MIB is {juniperMIB 1}. This branch of the MIB describes the Juniper Networks routers and their components, such as product line, product name, model, number of slots, and media space for holding Physical Interface Cards (PICs). It also provides information on the system's power supply state, board voltages, fans, temperatures, and air flow. In general, this branch of the structure of management information MIB is rarely polled for information because it is descriptive. However, you can poll this branch of the structure of management information MIB to determine the sysObjectId of a router as defined by MIB-II.

jnxServices

The object identifier for the jnxServices root branch is {juniperMIB 2}. The jnxServices root branch of the structure of management information MIB is a placeholder for future information.

jnxMibs

The object identifier for the jnxMIBs root branch is {juniperMIB 3} and includes one main subbranch, jnxBoxAnatomy, whose object identifier is {jnxMibs 1}. The other Juniper Networks enterprise-specific MIBs are also branches of jnxMibs. These Juniper Networks enterprise-specific MIBs include:

- MPLS MIB—Whose object identifier is {jnxMibs 2}.
- Juniper Networks enterprise-specific extensions to the interface MIB—Whose object identifier is {jnxMIBs 3}.
- Alarm MIB—Whose object identifier is {jnxMibs 4}.
- Firewalls MIB—Whose object identifier is {jnxMibs 5}.
- Destination class usage MIB—Whose object identifier is {jnxMibs 6}.
- Juniper Networks enterprise-specific extensions to the ping MIB—Whose object identifier is {jnxMibs 7}.
- Juniper Networks enterprise-specific extensions to the traceroute MIB—Whose object identifier is {jnxMibs 8}.
- ATM MIB—Whose object identifier is {jnxMibs 10}.
- IPv6 and ICMPv6 MIB—Whose object identifier is {jnxMibs 11}.
- IPv4 MIB—Whose object identifier is {jnxMibs 12}.
- Juniper Networks enterprise-specific extensions to the RMON MIB—Whose object identifier is {jnxMIBs 13}.
- Juniper Networks enterprise-specific extensions to the LDP traps MIB—Whose object identifier is {jnxMibs 14}.
- Class-of-service MIB—Whose object identifier is {jnxMibs 15}.
- Source class usage MIB—Whose object identifier is {jnxMibs 16}.
- Reverse-path-forwarding MIB—Whose object identifier is {jnxMibs 17}.
- Configuration management MIB—Whose object identifier is {jnxMibs 18}.
- Passive monitoring MIB—Whose object identifier is {jnxMibs 19}.
- SONET MIB—Whose object identifier is {jnxMibs 20}.
- ATM class-of-service MIB—Whose object identifier is {jnxMibs 21}.
- Ethernet MAC MIB—Whose object identifier is {jnxMibs 23}.
- SONET APS MIB—Whose object identifier is {jnxMibs 24}.
- Chassis defines MIB—Whose object identifier is {jnxMibs 25}.

- VPN MIB—Whose object identifier is {jnxMibs 26}.
- Flow collection services MIB—Whose object identifier is {jnxMibs 28}.
- RSVP Traffic Engineering (TE) MIB—Whose object identifier is {jnxMibs 30}.
- Host Resources MIB—Whose object identifier is {jnxMibs 31}.
- Services PIC MIB—Whose object identifier is {jnxMibs 32}.

For more information on these MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 105.

jnxTraps

The object identifier for the jnxTraps branch of the structure of management information MIB is {juniperMIB 4}. The jnxTraps branch contains the enterprise-specific SNMP traps supported by the JUNOS software. These Juniper Networks enterprise-specific SNMP traps include:

- jnxChassisTraps—Whose object identifier is {jnxTraps 1}.
- jnxChassisOKTraps—Whose object identifier is {jnxTraps 2}.
- jnxRmonTraps—Whose object identifier is {jnxTraps 3}.
- jnxLdpTraps—Whose object identifier is {jnxTraps 4}.
- jnxCmNotifications—Whose object identifier is {jnxTraps 5}.
- jnxSonetNotifications—Whose object identifier is {jnxTraps 6}.
- jnxPMonNotifications— Whose object identifier is {jnxTraps 7}
- jnxCollectorNotifications—Whose object identifier is {jnxTraps 8}.
- jnxPingNotificationPrefix—Whose object identifier is {jnxTraps 9}.
- jnxSpNotificationPrefix—Whose object identifier is {jnxTraps10}.

jnxExperiment

The object identifier for the jnxExperiment branch of the structure of management information MIB is {juniperMIB 5}. The jnxExperiment branch contains experimental Juniper Networks enterprise-specific MIBs. This is the top-level object identifier registry used by Juniper products for SNMP modules containing experimental MIB definitions.

jnxExperiment MIBs are defined as the following:

- IETF work-in-process MIBs that have not been assigned a permanent object identifier by the IANA.
- Juniper Networks work-in-process MIBs that have not achieved final production quality or field experience.

The following draft supports the jnxExperiment MIB space: Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version* (only jnxBgpM2PrefixInPrefixes, jnxBgpM2PrefixInPrefixesAccepted, and jnxBgpM2PrefixInPrefixesRejected objects).

Chapter 19

Interpreting the Enterprise-Specific Chassis MIBs

The enterprise-specific chassis MIB provides information on the router and its components. MIB objects represent each component and the status of the components. The enterprise-specific chassis definitions for the router model MIB contains the object identifiers (OIDs) that are used by the chassis MIB to identify platform and chassis components. The chassis MIB provides information that changes often. The chassis definitions for the router model MIB provides information that changes less often.

You can retrieve information from the MIB using any network management system (NMS). For a downloadable version of the chassis MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chassis.txt. For a downloadable version of the chassis definitions for the router model MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chas-defines.txt.

This chapter contains the following topics:

jnxBoxAnatomy

The object identifier for the jnxMIBs root branch is {juniperMIB 3} and includes one main subbranch, jnxBoxAnatomy, whose object identifier is {jnxMibs 1}.

The jnxBoxAnatomy MIB has the following sections:

- jnxBoxClass—See "Top-Level Objects" on page 245
- jnxBoxDescr—See "Top-Level Objects" on page 245
- jnxBoxSerialNo—See "Top-Level Objects" on page 245
- jnxBoxRevision—See "Top-Level Objects" on page 245
- jnxBoxInstalled—See "Top-Level Objects" on page 245
- jnxContainersTable on page 245
- jnxContentsLastChange on page 251
- jnxContentsTable on page 251
- jnxLEDLastChange on page 262
- jnxLEDTable on page 263
- jnxFilledLastChange on page 266
- jnxFilledTable on page 266
- jnxOperatingTable on page 274
- jnxRedundancyTable on page 283
- jnxFruTable on page 289

Top-Level Objects

The following branches of the jnxBoxAnatomy MIB are top-level objects:

- jnxBoxClass—The object identifier for the jnxBoxClass object is {jnxBoxAnatomy 1}. This object classifies the chassis product line.
- jnxBoxDescr—The object identifier for the jnxBoxDescr object is {jnxBoxAnatomy 2}. This object describes the chassis name and model.
- jnxBoxSerialNo—The object identifier for the jnxBoxSerialNo object is {jnxBoxAnatomy 3}. This object indicates the serial number of the chassis. jnxBoxSerialNo remains blank if the serial number is unknown or unavailable.
- jnxBoxRevision—The object identifier for the jnxBoxRevision object is {jnxBoxAnatomy 4}. This object indicates the last revision of the chassis.
- jnxBoxInstalled—The object identifier for the jnxBoxInstalled object is {jnxBoxAnatomy 5}. This object indicates the last time the box was installed and operational, represented by the sysUpTime value.

jnxContainersTable

The object identifier for the jnxContainersTable object is {jnxBoxAnatomy 6}. This object shows the structure of the chassis.

You can use the jnxContainersTable object to retrieve specific information on the router, such as how many of each component the router can contain. For example, the jnxContainersTable of an M20 router indicates that the router can accommodate four Flexible PIC Concentrators (FPCs); however, it does not describe how many FPCs the router actually has.

For more information on how many FPCs are actually on a router, see jnxContentsTable on page 251.

Entries within the jnxContainersTable object are represented by the jnxContainersEntry object, whose object identifier is {jnxContainersTable 1}. This jnxContainersEntry contains the following objects, which describe the contents of a particular router:

- jnxContainersIndex—The index value of an entry in the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 1}, which corresponds to jnxContainersType and jnxContainersDescr.
- jnxContainersView—The orientation of a container from the front of the router, whose object identifier is {jnxContainersEntry 2}. This object also indicates that the container is embedded in the router and how it is accessible from corresponding views. The value of this object is a bitmap represented as a sum. If multiple bits are set, you can access the container from that set of views. The values represent the bit positions and their corresponding views as follows:
  - 1—Front
  - 2—Rear

- 4—Top
- 8—Bottom
- 16—Left side
- 32—Right side

For each view plane, if specified counters are scattered in various views, the numbering sequence starts from left to right and then from top to bottom, as follows:

- Left side
- Right side
- Top
- Bottom
- Front
- Rear

NOTE: References to left and right sides are based on the view from the front of the chassis.

NOTE: In accordance with network management conventions, all indexes in the MIB begin with 1, not 0, although the slot number might be labeled 0.

- jnxContainersLevel—The abstraction level of the box or components for the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 3}. The level is enumerated from the outside to the inside, and from the outer layer to the inner layer.

  For example, if the top level (level 0) of the box refers to the chassis frame, then the next level (level 1) refers to the FPC slot within the chassis frame. Finally, the Physical Interface Card (PIC) space within the FPC slot of the chassis corresponds to level 2.

- jnxContainersWithin—The container housing the entry at the next-higher level of the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 4}.

  For example, the within value for jnxMediaCardSpacePIC.0 is 7. Because the jnxM20SlotFPC.0 retains an index value of 7, the FPC houses the PIC.

- jnxContainersType—The component of the chassis MIB at a specific index, view, level, and within value for the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 5}.

- jnxContainersDescr—The description of the component in the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 6}.

- jnxContainersCount—The maximum number of a given component that the router can accommodate within the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 7}.

  For example, the M20 router can house a specific maximum number of FPCs within the chassis frame. The maximum number is not necessarily the actual number of FPCs; this can change dynamically.

Table 28 through Table 35 provide examples of jnxContainersEntry objects in the jnxContainersTable. The following column headings for each table are abbreviated to correspond to the parts of the jnxContainersEntry objects:

- Index—jnxContainersIndex
- View—jnxContainersView
- Level—jnxContainersLevel
- Within—jnxContainersWithin
- Type—jnxContainersType
- Description—jnxContainersDescr
- Count—jnxContainersCount

Table 28 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M40 router.

Table 28: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM40.0	Chassis frame compartment	1
2	2	1	1	jnxSlotPowerSupply.0	Power supply compartment	2
3	3	1	1	jnxSlotCoolingImpeller.0	Impeller compartment	2
4	2	1	1	jnxSlotCoolingFan.0	Fan compartment	3
5	2	1	1	jnxSlotHostCtrl.0	Host controller compartment	1
6	1	1	1	jnxSlotSCB.0	SCB slot	1
7	1	1	1	jnxSlotFPC.0	FPC slot	8
8	1	2	7	jnxMediaSlotCardPIC.0	PIC space	4
9	2	1	1	jnxSlotRoutingEngine.0	Routing Engine compartment	1

Table 29 describes objects in the jnxContainersTable of an M20 router.

Table 29: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM20.0	Chassis frame compartment	1
2	2	1	1	jnxM20SlotPower.0	Power supply compartment	2
4	3	1	1	jnxSlotFan.0	Fan compartment	4
6	2	1	1	jnxM20SlotSSB.0	SSB slot	2
7	1	1	1	jnxM20SlotFPC.0	FPC slot	4
8	1	2	7	jnxM20MediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM20RE.0	Routing Engine compartment	2
10	1	1	1	JNXM20FrontPanel.0	Front display slot	1

Table 30 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M160 router.

Table 30: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM160.0	Chassis frame compartment	1
2	2	1	1	Jnx160SlotPower.0	Power supply compartment	2
4	3	1	1	jnxM160SlotFan.0	Fan compartment	4
6	2	1	1	jnxM160SlotSFM.0	SFM slot	4
7	1	1	1	jnxM160SlotFPC.0	FPC slot	8
8	1	2	7	jnxM160MediaCardSlotPIC.0	PIC space	4
9	2	1	1	jnxM160SlotHM.0	Host slot	2
10	1	1	1	jnxM160SlotFPM.0	FPM slot	1
11	2	1	1	jnxM160SlotPCG.0	PCG slot	2
12	2	1	1	jnxM160SlotMCS.0	MCS slot	2
13	1	1	1	jnxM160SlotCIP.0	CIP slot	1

Table 31 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M10 router.

Table 31: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM10.0	Chassis frame compartment	1
2	2	1	1	jnxM10SlotPower.0	Power supply compartment	2
4	2	1	1	jnxM10SlotFan.0	Fan compartment	1
6	2	1	1	jnxM10SlotFEB.0	FEB slot	1
7	1	1	1	jnxM10SlotFPC.0	FPC slot	2
8	1	2	7	jnxM10MediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM10SlotRE.0	Routing Engine compartment	1

Table 32 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M5 router.

Table 32: jnxContainersEntry Objects in the jnxContainersTable of an M5 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM5.0	Chassis frame compartment	1
2	2	1	1	jnxM5SlotPower.0	Power supply compartment	2
4	3	1	1	jnxM5SlotFan.0	Fan compartment	4
6	2	1	1	jnxM5SlotFEB.0	FEB slot	1
7	1	1	1	jnxM5SlotFPC.0	FPC slot	1
8	1	2	7	jnxM5MediaCardSlotPIC.0	PIC space	4
9	2	1	1	jnxM5SlotRE.0	Routing Engine compartment	1

Table 33 describes objects contained in a jnxContainersEntry in the jnxContainersTable of a T640 routing node.

Table 33: jnxContainersEntry Objects in the jnxContainersTable of a T640 Routing Node

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisT640.0	Chassis frame	1
2	2	1	1	jnxT640SlotPower.0	PEM slot	2
4	3	1	1	jnxT640SlotFan.0	Fan slot	3
7	1	1	1	jnxT640SlotFPC.0	FPC slot	8
8	1	2	7	jnxT640MediaCardSpacePIC.0	PIC slot	4
9	2	1	1	jnxT640SlotHM.0	Host slot	2
10	1	1	1	jnxT640SlotFPB.0	FPM slot	1
11	2	1	1	jnxT640SlotSCG.0	SCG slot	2
12	2	1	1	jnxT640SlotCB.0	CG slot	2
13	1	1	1	jnxT640SlotCIP.0	CIP slot	1
14	2	1	1	jnxT640SlotSPMB.0	SPMB slot	2
15	2	1	1	jnxT640SlotSIB.0	SIB slot	5

Table 34 describes objects contained in a jnxContainersEntry in the jnxContainersTable of a T320 router.

Table 34: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisT320.0	Chassis frame	1
2	2	1	1	jnxT320SlotPower.0	PEM slot	2
4	3	1	1	jnx320SlotFan.0	Fan slot	3
7	1	1	1	jnxT320SlotFPC.0	FPC slot	8
8	1	2	7	jnxT320MediaCardSpacePIC.0	PIC slot	2
9	2	1	1	jnxT320SlotHM.0	Host slot	2
10	1	1	1	jnxT320SlotFPB.0	FPM slot	1
11	2	1	1	jnxT320SlotSCG.0	SCG slot	2
12	2	1	1	jnxT320SlotCB.0	CB slot	2
13	1	1	1	jnxT320SlotCIP.0	CIP slot	1
14	2	1	1	jnxT320SlotSPMB.0	SPMB slot	2
15	2	1	1	jnxT320SlotSIB.0	SIB slot	3

Table 35 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M40e router.

Table 35: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM40e.0	Chassis frame compartment	1
2	2	1	1	jnxM40eSlotPower.0	Power supply compartment	2
4	3	1	1	jnxM40eSlotFan.0	Fan compartment	4
6	2	1	1	jnxM40eSlotSFM.0	SFM slot	2
7	1	1	1	jnxM40eSlotFPC.0	FPC slot	8
8	1	2	7	jnxM40eMediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM40eSlotHM.0	Host slot	2
10	1	1	1	jnxM40eSlotFPM.0	FPM slot	1
11	2	1	1	jnxM40eSlotPCG.0	PCG slot	2
12	2	1	1	jnxM40eSlotMCS.0	MCS slot	2
13	1	1	1	jnxM40eSlotCIP.0	CIP slot	1

jnxContentsLastChange

The object identifier for jnxContentsLastChange object is {jnxBoxAnatomy 7}. This object indicates the time at which the box contents last changed, represented by the sysUpTime value.

jnxContentsTable

The object identifier for jnxContentsTable object is {jnxBoxAnatomy 8}. This object specifies the contents of the chassis.

The jnxContentsTable lists the contents of an entry, which are defined as follows:

- jnxContentsContainerIndex—Associates the jnxContainersIndex with the jnxContainersTable, whose object identifier is {jnxContentsEntry 1}.
- jnxContentsL1Index—The level-one index of the container housing the component, whose object identifier is {jnxContentsEntry 2}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.

NOTE: MIBs start with a value of 1, while the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- jnxContentsL2Index—The level-two index of the container housing the component, whose object identifier is {jnxContentsEntry 3}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.

- jnxContentsL3Index—The level-three index of the container housing the component, whose object identifier is {jnxContentsEntry 4}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxContentsType—The component at a specific container index or L1, L2, or L3 index, whose object identifier is {jnxContentsEntry 5}.
- jnxContentsDescr—The type of component described in plain English, whose object identifier is {jnxContentsEntry 6}.
- jnxContentsSerialNo—The serial number of the component, whose object identifier is {jnxContentsEntry 7}.
- jnxContentsRevision—The revision level of the component, whose object identifier is {jnxContentsEntry 8}.
- jnxContentsInstalled—The time at which the component was last installed and operational, represented by the sysUpTime value, whose object identifier is {jnxContentsEntry 9}.
- jnxContentsPartNo—The part number of the component (blank if unknown or unavailable), whose object identifier is {jnxContentsEntry 10}.

Table 36 through Table 38 provide examples of jnxContentEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxContentsEntry objects:

- Container index— jnxContentsContainerIndex
- L1—jnxContentsL1Index
- L2—jnxContentsL2Index
- L3—jnxContentsL3Index
- Type—jnxContentsType
- Description—jnxContentsDescr
- Serial Number—jnxContentsSerialNo
- Revision—jnxContentsRevision
- Installed—jnxContentsInstalled
- Part Number—jnxContentsPartNo

Table 36 provides an example of jnxContentEntry objects in the jnxContentTable of an M20 router.

Table 36: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	1	0	jnxBackplaneM20.0	Midplane	AL3280	REV07	0:0:00:00.00	710-00157
2	1	0	0	jnxM20PowerDC.0	DC power supply A	001652	REV 05	0:0:00:00.00	740-00146
2	2	0	0	jnxM20PowerDC.0	DC power supply B	001652	REV 05	0:0:00:00.00	740-00146
4	1	0	0	jnxM20Fan.0	Front top fan			0:0:00:00.00	
4	2	0	0	jnxM20Fan	Middle fan			0:0:00:00.00	
4	3	0	0	jnxM20Fan	Bottom fan			0:0:00:00.00	
4	4	4	0	jnxM20Fan	Rear fan			0:0:00:00.00	
6	1	0	0	jnxM20SSB.0	SSB 0 Internet Processor II	AG0809	REV 01	0:0:00:35.17	710-001951
7	1	0	0	jnxM20FPC.0	FPC @ 0/*/*	AN1335	REV 01	0:0:01:01.80	710-001292
7	2	0	0	jnxM20FPC.0	FPC @ 1/*/*	AN1124	REV 01	0:0:01:07:96	710-001292
7	3	0	0	jnxM20FPC.0	FPC @ 2/*/*	AN1726	REV 01	0:0:01:14:12	710-001292
7	4	0	0	jnxM20FPC.0	FPC @ 3/*/*	AN1691	REV 01	0:0:01:20.28	710-001292
8	1	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/0/*	HD4313	REV 04	0:0:00:00.00	750-002992
8	1	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/1/*	AJ5844	REV 04	0:0:00:00.00	750-002992
8	1	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/2/*	HD4518	REV 04	0:0:00:00.00	750-002992
8	1	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/3/*	HD4515	REV 04	0:0:00:00.00	750-002992
8	2	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/0/*	HD4296	REV 04	0:0:00:00.00	750-002992
8	2	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/1/*	HD4323	REV 04	0:0:00:00.00	750-002992
8	2	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/2/*	HD4129	REV 04	0:0:00:00.00	750-002992
8	2	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/3/*	HD4341	REV 04	0:0:00:00.00	750-002992
8	3	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/0/*	AH4147	REV 07	0:0:00:00.00	750-002303

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	3	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/1/*	AH4238	REV 07	0:0:00:00.00	750-002303
8	3	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/2/*	AH4116	REV 07	0:0:00:00.00	750-002303
8	3	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/3/*	AH4208	REV 07	0:0:00:00.00	750-002303
8	4	1	0	jnxM20GigEther.0	PIC: 1x G/E, 1000BASE-SX @ 3/0/*	AS3697	REV 07	0:0:00:00.00	750-001072
8	4	2	0	jnxM20ChOc12toDS3.0	PIC: 1x COC12SMIR @ 3/1/*	AE1110	REV 08	0:0:00:00.00	750-001190
8	4	4	0	jnxM20ChStml1.0	PIC: 1x CSTM1SMIR @ 3/3/*	AD9599	REV 04	0:0:00:00.00	750-003250
9	1	0	0	jnxM20RE.0	Routing Engine			3:16:16:53.21	
10	1	0	0	jnxM20FrontPanel.0	Front panel display			0:0:00:00.00	

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M20 router is listed below.

```
user@host> show chassis hardware
Item            Version Part number Serial Number   Description
Chassis53711    M20
Backplane       REV 07  710-001517      AL3280
Power Supply A  REV 05  740-001466      001652      DC
Power Supply B  REV 05  740-001466      001632      DC
Display         REV 04  710-001519      AP9225
Host 0 c900000619e6ba01 teknor
SSB slot 0      REV 01  710-001951      AG0809      Internet Processor
II
FPC 0      REV 01  710-001292  AN1335
 PIC 0     REV 04  750-002992  HD4313      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  AJ5844      4x F/E, 100 BASE-TX
 PIC 2     REV 04  750-002992  HD4518      4x F/E, 100 BASE-TX
 PIC 3     REV 04  750-002992  HD4515      4x F/E, 100 BASE-TX
FPC 1      REV 01  710-001292  AN1124
 PIC 0     REV 04  750-002992  HD4296      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  HD4323      4x F/E, 100 BASE-TX
 PIC 2     REV 04  750-002992  HD4129      4x F/E, 100 BASE-TX
 PIC 3     REV 04  750-002992  HD4341      4x F/E, 100 BASE-TX
FPC 2      REV 01  710-001292  AN1726
 PIC 0     REV 07  750-002303  AH4147      4x F/E, 100 BASE-TX
 PIC 1     REV 07  750-002303  AH4238      4x F/E, 100 BASE-TX
 PIC 2     REV 07  750-002303  AH4116      4x F/E, 100 BASE-TX
 PIC 3     REV 07  750-002303  AH4208      4x F/E, 100 BASE-TX
FPC 3      REV 01  710-001292  AN1691
 PIC 0     REV 08  750-001072  AS3697      1x G/E, 1000
BASE-SX
 PIC 1     REV 03  750-001190  AE1110      1x COC12, SMIR
 PIC 3     REV 04  750-003250  AD9599      1x CSTM1, SMIR
```

Table 37 provides an example of jnxContentEntry objects in the jnxContentTable of a T640 routing node.

Table 37: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxMidplaneT640.0	Midplane	AX5633	REV 04	0:0:00:00.00	710-002726
2	2	0	0	jnxT640Power.0	PEM 1	MD21815	RevX02	0:0:00:00.00	740-002595
4	1	1	0	jnxT640Fan.0	Top left front fan			0:0:00:00.00	
4	1	2	0	jnxT640Fan.0	Top left middle fan			0:0:00:00.00	
4	1	3	0	jnxT640Fan.0	Top left rear fan			0:0:00:00.00	
4	1	4	0	jnxT640Fan.0	Top right front fan			0:0:00:00.00	
4	1	5	0	jnxT640Fan.0	Top right middle fan			0:0:00:00.00	
4	1	6	0	jnxT640Fan.0	Top right rear fan			0:0:00:00.00	
4	2	1	0	jnxT640Fan.0	Bottom left front fan			0:0:00:00.00	
4	2	2	0	jnxT640Fan.0	Bottom left middle fan			0:0:00:00.00	
4	2	3	0	jnxT640Fan.0	Bottom left rear fan			0:0:00:00.00	
4	2	4	0	jnxT640Fan.0	Bottom right front fan			0:0:00:00.00	
4	2	5	0	jnxT640Fan.0	Bottom right middle fan			0:0:00:00.00	
4	2	6	0	jnxT640Fan.0	Bottom right rear fan			0:0:00:00.00	
4	3	1	0	jnxT640Fan.0	Fourth blower from top			0:0:00:00.00	
4	3	2	0	jnxT640Fan.0	Bottom blower			0:0:00:00.00	
4	3	3	0	jnxT640Fan.0	Middle blower			0:0:00:00.00	
4	3	4	0	jnxT640Fan.0	Top blower			0:0:00:00.00	
4	3	5	0	jnxT640Fan.0	Second blower from top			0:0:00:00.00	
7	2	0	0	jnxT640FPC.0	FPC @ 1/*/*	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	1	0	jnxT640FPC.0	FPC @ 1/0/* top temp. sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	2	0	jnxT640FPC.0	FPC @ 1/1/* bottom temp. sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	6	0	0	jnxT640FPC.0	FPC @ 5/*/*	HD5001	REV 03	0:18:57:02.71	710-001721

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	6	1	0	jnxT640FPC.0	FPC @ 5/0/* top temp. sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	6	2	0	jnxT640FPC.0	FPC @ 5/1/* bottom temp. sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	8	0	0	jnxT640FPC.0	FPC @ 7/*/*	HE3179	REV 01	0:18:56:52.85	710-002385
7	8	1	0	jnxT640FPC.0	FPC @ 7/0/* top temp. sensor	HE3179	REV 01	0:18:56:52.85	710-002385
7	8	2	0	jnxT640FPC.0	FPC @ 7/1/* bottom temp. sensor	HE3179	REV 01	0:18:56:52.85	710-002385
8	2	1	0	jnxT640PIC3.0	PIC: 1x G/E, 1000 BASE-SX @ 1/0/*	AP5542	REV 08	0:18:56:50.91	750-001072
8	2	2	0	jnxT640PIC3.0	PIC: 1x OC-12 ATM, SMIR @ 1/1/*	AK6894	REV 02	0:18:56:55.24	750-002983
8	2	3	0	jnxT640PIC3.0	PIC: 1x G/E, 1000 BASE-SX @ 1/2/*	HD4968	REV 04	0:18:56:55.64	750-001894
8	6	1	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/0/*	HC0273	REV 01	0:18:57:04.47	750-004535
8	6	2	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/1/*	HC0271	REV 01	0:18:57:04.55	750-004535
8	6	3	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/2/*	HC0254	REV 01	0:18:57:04.64	750-004535
8	8	1	0	jnxT640PIC3.0	PIC: 2x G/E, 1000 BASE-SX @ 7/0/*	AD3632	REV 01	0:18:56:55.16	710-002381
8	8	2	0	jnxT640PIC3.0	PIC: 4x OC-12 SONET, SMIR @ 7/1/*	AD3831	REV 05	0:18:56:55.18	750-001901
8	8	3	0	jnxT640PIC3.0	PIC: 1x OC-48 SONET, SMIR @ 7/2/*	AA9603	REV 01	0:18:56:55.21	750-001900
8	8	4	0	jnxT640PIC3.0	PIC: 1x OC-48 SONET, SMSR @ 7/3/*	AD5724	REV 05	0:18:56:55.24	750-001900

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
9	1	0	0	jnxT640HM.0	Host 0			0:19:19:30.95	
9	2	0	0	jnxT640HM.0	Host 1	210865700292	REV 01	2:19:45:51.00	740-005022
10	1	0	0	jnxT640FPB.0	FPM	HE3245	REV 02	0:0:00:00.00	710-002901
11	1	0	0	jnxT640SCG.0	SCG 0	HF6023	REV 04	0:0:00:00.00	710-003423
11	2	0	0	jnxT640SCG.0	SCG 1	HF6061	REV 04	0:0:00:00.00	710-003423
12	2	0	0	jnxT640CB.0	CB 0	HE3614	REV 06	0:0:00:00.00	710-002728
12	2	0	0	jnxT640CB.0	CB 1	HE3627	REV 06	0:0:00:00.00	710-002728
13	1	0	0	jnxT640CIP.0	CIP	HA4729	REV 05	0:0:00:00.00	710-002895
14	1	0	0	jnxT640SPMB.0	SPMB 0	HF6876	REV 02	0:18:56:06.72	710-003229
14	2	0	0	jnxT640SPMB.0	SPMB 1	HG6237	REV 02	0:18:56:08.01	710-003229
15	1	0	0	jnxT640SIB.0	SIB 0	HJ9669	REV 02	0:0:00:00.00	710-005157
15	2	0	0	jnxT640SIB.0	SIB 1	HJ9668	REV 02	0:0:00:00.00	710-005157
15	3	0	0	jnxT640SIB.0	SIB 2	HH3039	REV 02	0:0:00:00.00	710-005157
15	4	0	0	jnxT640SIB.0	SIB 3	HH3041	REV 02	0:0:00:00.00	710-005157
15	5	0	0	jnxT640SIB.0	SIB 4	HJ9657	REV 02	0:0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item       Version Part number Serial number   Description
Chassis     T640
Midplane    REV 04  710-002726  AX5633
FPM GBUS    REV 02  710-002901  HE3245
FPM Display REV 02  710-002897  HA4873
CIP         REV 05  710-002895  HA4729
PEM 1       RevX02  740-002595  MD21815     Power Entry Module
SCG 0       REV 04  710-003423  HF6023
SCG 1       REV 04  710-003423  HF6061
Host 0      unknown
Host 1      REV 01  740-005022  210865700292   RE-3.0
CB 0        REV 06  710-002728  HE3614
CB 1        REV 06  710-002728  HE3627
FPC 1       REV 01  710-002385  HE3009      FPC Type 1
 CPU        REV 06  710-001726  HC0010
 PIC 0      REV 08  750-001072  AP5542      1x G/E, 1000 BASE-SX
 PIC 1      REV 02  750-002983  AK6894      1x OC-12 ATM, SMIR
 PIC 2      REV 04  750-001894  HD4968      1x G/E, 1000 BASE-SX
 MMB 1      REV 03  710-001723  HE7264      MMB-144mbit
 ICBM       REV 01  710-003384  HE3042
 PPB 0      REV 01  710-003758  HE7173      PPB Type 2
 PPB 1      REV 01  710-003758  HE7170      PPB Type 2
FPC 5       REV 03  710-001721  HD5001      FPC Type 3
 CPU        REV 06  710-001726           HA5080
 PIC 0      REV 01  750-004535  HC0273      1x OC-192 SM SR1
 PIC 1      REV 01  750-004535  HC0271      1x OC-192 SM SR1
 PIC 2      REV 01  750-004535  HC0254      1x OC-192 SM SR1
 MMB 0      REV 03  710-001723  HE7263      MMB-144mbit
 MMB 1      REV 03  710-001723  HE7266      MMB-144mbit
 ICBM       REV 01  710-003384  HE3044
 PPB 0      REV 02  710-002845  HD6027      PPB Type 3
 PPB 1      REV 02  710-002845  HD6039      PPB Type 3
FPC 7       REV 01  710-002385  HE3179      FPC Type 2
 CPU        REV 06  710-001726           HE7915
 PIC 0      REV 01  710-002381  AD3632      2x G/E, 1000 BASE-SX
 PIC 1      REV 05  750-001901  AD3831      4x OC-12 SONET, SMIR
 PIC 2      REV 01  750-001900  AA9603      1x OC-48 SONET, SMIR
 PIC 3      REV 05  750-001900  AD5724      1x OC-48 SONET, SMSR
 MMB 1      REV 02  710-004047  HE3424      MMB-288mbit
 ICBM       REV 04  710-003384  HA4480
 PPB 0      REV 02  710-003758  HE3169      PPB Type 2
 PPB 1      REV 02  710-003758  HA4535      PPB Type 2
SPMB 0      REV 02  710-003229 HF6876
SPMB 1      REV 02  710-003229 HG6237
SIB 0       REV 02  710-005157  HJ9669      SIB-I8-F16
SIB 1       REV 02  710-005157  HJ9668      SIB-I8-F16
SIB 2       REV 02  710-005157  HH3039      SIB-I8-F16
SIB 3       REV 02  710-005157  HH3041      SIB-I8-F16
SIB 4       REV 02  710-005157  HJ9657      SIB-I8-F16
```

Table 38 provides an example of jnxContentEntry objects in the jnxContentTable of a T320 router.

Table 38: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxMidplaneT320.0	Midplane	AY4527	Rev 01	(0) 0:00:00.00	710-004339
2	1	0	0	jnxT320Power.0	PEM 0	ML14099	Rev 01	(0) 0:00:00.00	
4	1	1	0	jnxT320Fan.0	Top left front fan			(0) 0:00:00.00	
4	1	2	0	jnxT320Fan.0	Top left middle fan			(0) 0:00:00.00	
4	1	3	0	jnxT320Fan.0	Top left rear fan			(0) 0:00:00.00	
4	1	4	0	jnxT320Fan.0	Top right front fan			(0) 0:00:00.00	
4	1	5	0	jnxT320Fan.0	Top right middle fan			(0) 0:00:00.00	
4	1	6	0	jnxT320Fan.0	Top right rear fan			(0) 0:00:00.00	
4	2	1	0	jnxT320Fan.0	Bottom left front fan			(0) 0:00:00.00	
4	2	2	0	jnxT320Fan.0	Bottom left middle fan			(0) 0:00:00.00	
4	2	3	0	jnxT320Fan.0	Bottom left rear fan			(0) 0:00:00.00	
4	2	4	0	jnxT320Fan.0	Bottom right front fan			(0) 0:00:00.00	
4	2	5	0	jnxT320Fan.0	Bottom right middle fan			(0) 0:00:00.00	
4	2	6	0	jnxT320Fan.0	Bottom right rear fan			(0) 0:00:00.00	
4	3	1	0	jnxT320Fan.0	Rear tray top fan			(0) 0:00:00.00	
4	3	2	0	jnxT320Fan.0	Rear tray second fan			(0) 0:00:00.00	
4	3	3	0	jnxT320Fan.0	Rear tray middle fan			(0) 0:00:00.00	
4	3	4	0	jnxT320Fan.0	Rear tray fourth fan			(0) 0:00:00.00	
4	3	5	0	jnxT320Fan.0	Rear tray bottom fan			(0) 0:00:00.00	
7	4	0	0	jnxT320FPC.0	FPC @ 3/*/*	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	1	0	jnxT320FPC.0	FPC @ 3/0/* top temp. sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	2	0	jnxT320FPC.0	FPC @ 3/1/* bottom temp. sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	1	1	0	jnxT320PIC3	PIC: 1x OC-192 SM SR2 @ 0/0/*	HJ9283	REV 06	(6378) 0:01:03.78	750-004535
8	1	2	0	jnxT320PIC3	PIC: 1x OC-192 SM SR2 @ 0/1/*	HJ9298	REV 06	(6434) 0:01:04.34	750-004535
9	1	0	0	jnxT320HM.0	Host 0	2108657 00286	REV 01	(32762924) 3 days, 19:00:29.24	740-005022
9	2	0	0	jnxT320HM.0	Host 1	2109290 00186	REV 01	(110269900) 12 days, 18:18:19.00	740-005022
10	1	0	0	jnxT320FPB.0	FPM	AY4514	REV 02	(0) 0:00:00.00	710-004461
11	1	0	0	jnxT320SCG.0	SCG 0	AY4520	REV 06	(0) 0:00:00.00	710-004455
11	2	0	0	jnxT320SCG.0	SCG 1	AY4526	REV 06	(0) 0:00:00.00	710-004455
12	1	0	0	jnxT320CB.0	CB 0	AY4765	REV 11	(0) 0:00:00.00	710-002728
12	2	0	0	jnxT320CB.0	CB 1	HG6051	REV 06	(0) 0:00:00.00	710-002728
13	1	0	0	jnxT320CIP.0	CIP	HC0476	REV 05	(0) 0:00:00.00	710-002895
14	1	0	0	jnxT320SPMB.0	SPMB 0	HB1893	REV 02	(26186997) 3 days, 0:44:29.97	710-003229
14	2	0	0	jnxT320SPMB.0	SPMB 1	HD5520	REV 02	(26186913) 3 days, 0:44:29.13	710-003229
15	1	0	0	jnxT320SIB.0	SIB 0	BC1509	REV 02	(0) 0:00:00.00	710-005157
15	2	0	0	jnxT320SIB.0	SIB 1	BC1512	REV 02	(0) 0:00:00.00	710-005157
15	3	0	0	jnxT320SIB.0	SIB 2	BC1494	REV 02	(0) 0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T320 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis  T320
Midplane         REV 01   710-004339   AY4527
FPM GBUS         REV 02   710-004461   AY4514
FPM Display      REV 02   710-002897   HF6097
CIP              REV 05   710-002895   HC0476
PEM 0            Rev 01   740-004359   ML14099           Power Entry
Module
SCG 0            REV 06   710-004455   AY4520
SCG 1            REV 06   710-004455   AY4526
RE 0            REV 01   740-005022   210865700286      RE-3.0
RE 1            REV 01   740-005022   210929000186      RE-3.0
CB 0             REV 11   710-002728   AY4765
CB 1             REV 06   710-002728   HG6051
FPC 1            REV 01   710-004333   AY4507            FPC Type 3
  CPU            REV 06   710-001726   HA4719
  MMB 1          REV 03   710-004047   HD5738            MMB-288mbit
  PPB 0          REV 02   710-002845   HC0988            PPB Type 3
FPC 3            REV 01   710-004333   AY4706            FPC Type 3
  CPU            REV 06   710-001726   HE7916
  MMB 1          REV 03   710-004047   HG6326            MMB-288mbit
  PPB 0          REV 02   710-002845   HC0958            PPB Type 3
SPMB 0           REV 02   710-003229   HB1893
SPMB 1           REV 02   710-003229   HD5520
SIB 0            REV 02   710-005157   BC1509            SIB-I8-F16
SIB 1            REV 02   710-005157   BC1512            SIB-I8-F16
SIB 2            REV 02   710-005157   BC1494            SIB-I8-F16
```

jnxLEDLastChange

The object identifier for the jnxLEDLastChange object is {jnxBoxAnatomy 9}. This object indicates when the LED last changed state. Its value is 0 if the sysUpTime value is unknown, or if it already existed when the agent was active.

jnxLEDTable

The object identifier for the jnxLEDTable object is {jnxBoxAnatomy 10}. This object indicates the LED status of the router and lists the contents of an entry. Entries in the jnxLEDTable are represented by the jnxLEDEntry object, whose object identifier is {jnxLEDTable 1}.

The jnxLEDTable describes the components of the LED Box Indicators, whose elements are described as follows:

- jnxLEDAssociateTable—The associate table to which the entry is related, whose object identifier is {jnxLEDEntry 1}.
- jnxLEDAssociateIndex—The index of the subject in the associated table to which the entry is related, whose object identifier is {jnxLEDEntry 2}. The associate index is the index of the subject in the associated table, which returns you to the jnxContainersTable.
- jnxLEDL1Index—The level-one index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 3}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.

NOTE: MIBs start with a value of 1, while the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- jnxLEDL2Index—The level-two index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 4}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxLEDL3Index—The level-three index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 5}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxLEDOriginator—The chassis component that originated the update, whose object identifier is {jnxLEDEntry 6}.
- jnxLEDDescr—The name or detailed description of the entry, whose object identifier is {jnxLEDEntry 7}.

- jnxLEDState—The state of the LED indicator, whose object identifier is {jnxLEDEntry 8}. The state can be any of the following:
  - Amber—Alarm, offline, not working
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Other—Unknown or unavailable
  - Red—Alert, component failed
  - Yellow—Alarm, warning
- jnxLEDStateOrdered—The state of the LED indicator, whose object identifier is {jnxLEDEntry 9}. jnxLEDStateOrdered provides the same information as jnxLEDState but lists the states in a different order. The state can be any of the following:
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Amber—Alarm, offline, not working
  - Yellow—Alarm, warning
  - Red—Alert, component failed
  - Other—Unknown or unavailable

Table 39 through Table 41 provide examples of jnxLEDEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxLEDEntry objects:

- Associate table—jnxLEDAssociateTable
- L1—jnxLEDL1Index
- L2—jnxLEDL2Index
- L3—jnxLEDL3Index
- Originator—jnxLEDOriginator
- Description—jnxLEDDescr
- State—jnxLEDState

Table 39 provides an example of jnxLEDEntry objects in the jnxLEDTable of an M20 router.

Table 39: jnxLEDEntry Objects in the jnxLEDTable of an M20 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	jnxChassisM20.0	Chassis alarm LED	Other
jnxContentsTable	6	1	0	0	jnxM20SSB.0	SSB 1 LED	Blue
jnxContentsTable	6	2	0	0	jnxM20SSB.0	SSB 2 LED	Green
jnxContentsTable	7	1	0	0	jnxM20FPC.0	FPC 1 LED	Amber
jnxContentsTable	7	2	0	0	jnxM20FPC.0	FPC 2 LED	Blue
jnxContentsTable	7	3	0	0	jnxM20FPC.0	FPC 3 LED	Blue
jnxContentsTable	7	4	0	0	jnxM20FPC.0	FPC 4 LED	Amber
jnxContentsTable	9	1	0	0	jnxM20RE.0	Routing Engine 1 LED	Blue
jnxContentsTable	9	2	0	0	jnxM20RE.0	Routing Engine 2 LED	Other

Table 40 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T640 routing node.

Table 40: jnxLEDEntry Objects in the jnxLEDTable of a T640 Routing Node

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	jnxChassisT640.0	Chassis alarm LED	Other
jnxContentsTable	7	1	0	0	jnxT640FPC.0	FPC slot 0 LED	Other
jnxContentsTable	7	2	0	0	jnxT640FPC.0	FPC slot 1 LED	Green
jnxContentsTable	7	3	0	0	jnxT640FPC.0	FPC slot 2 LED	Other
jnxContentsTable	7	4	0	0	jnxT640FPC.0	FPC slot 3 LED	Other
jnxContentsTable	7	5	0	0	jnxT640FPC.0	FPC slot 4 LED	Other
jnxContentsTable	7	6	0	0	jnxT640FPC.0	FPC slot 5 LED	Green
jnxContentsTable	7	7	0	0	jnxT640FPC.0	FPC slot 6 LED	Other
jnxContentsTable	7	8	0	0	jnxT640FPC.0	FPC slot 7 LED	Green
jnxContentsTable	9	1	0	0	jnxT640HM.0	Host 0 LED	Blue
jnxContentsTable	9	2	0	0	jnxT640HM.0	Host 1 LED	Green

Table 41 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T320 router.

Table 41: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable(3)	1	1	0	0	jnxChassisT320.0	Chassis alarm LED	Other
jnxContentsTable(3)	7	1	0	0	jnxT320FPC.0	FPC slot 0 LED	Other
jnxContentsTable(3)	7	2	0	0	jnxT320FPC.0	FPC slot 1 LED	Other
jnxContentsTable(3)	7	3	0	0	jnxT320FPC.0	FPC slot 2 LED	Other
jnxContentsTable(3)	7	4	0	0	jnxT320FPC.0	FPC slot 3 LED	Other
jnxContentsTable(3)	7	5	0	0	jnxT320FPC.0	FPC slot 4 LED	Other
jnxContentsTable(3)	7	6	0	0	jnxT320FPC.0	FPC slot 5 LED	Other
jnxContentsTable(3)	7	7	0	0	jnxT320FPC.0	FPC slot 6 LED	Other
jnxContentsTable(3)	7	8	0	0	jnxT320FPC.0	FPC slot 7 LED	Other
jnxContentsTable(3)	9	1	0	0	jnxT320HM.0	Host 0 LED	Blue
jnxContentsTable(3)	9	2	0	0	jnxT320HM.0	Host 1 LED	Green

jnxFilledLastChange

The object identifier for the jnxFilledLastChange object is {jnxBoxAnatomy 11}. This object indicates when the box filled status last changed. This variable is 0 if the sysUpTime value is unknown or it already existed when the agent was active.

jnxFilledTable

The object identifier for the jnxFilledTable object is {jnxBoxAnatomy 12}. This object indicates whether a specific container in the router is used (filled) or empty. This table is used for inventory and capacity planning.

Entries in the jnxFilledTable are represented by the jnxFilledEntry object, whose object identifier is {jnxFilledTable 1}.

The jnxFilledTable describes the status of specific containers whose component objects are described as follows:

- jnxFilledContainerIndex—The associated jnxContainersIndex in the jnxContainersTable, whose object identifier is {jnxFilledEntry 1}.
- jnxFilledL1Index—The level-one index of the container housing the entry, whose object identifier is {jnxFilledEntry 2}.
- jnxFilledL2Index—The level-two index of the container housing the entry, whose object identifier is {jnxFilledEntry 3}.
- jnxFilledL3Index—The level-three index of the container housing the entry, whose object identifier is {jnxFilledEntry 4}.

- jnxFilledDescr—The entry's name or detailed description of the entry, whose object identifier is {jnxFilledEntry 5}.
- jnxFilledState—The entry's state (filled or empty), whose object identifier is {jnxFilledEntry 6}.

Table 42 through Table 44 provide examples of jnxFilledEntry objects in the jnxFilledTable. The following column headings for each table are abbreviated to correspond to the parts of the jnxFilledEntry objects:

- Container index—jnxFilledContainerIndex
- L1—jnxFilledL1Index
- L2—jnxFilledL2Index
- L3—jnxFilledL3Index
- Description—jnxFilledDescr
- State—jnxFilledState

Table 42 provides an example of jnxFilledEntry objects in the jnxFilledTable of an M20 router.

Table 42: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame compartment	Filled
1	1	1	0	Temperature sensor space 0	Filled
1	1	2	0	Temperature sensor space 1	Filled
2	1	0	0	Power supply compartment A	Filled
2	2	0	0	Power supply compartment B	Empty
3	1	0	0	Rear top impeller compartment	Filled
3	2	0	0	Front bottom impeller compartment	Filled
4	1	0	0	Rear left fan compartment	Filled
4	2	0	0	Right center fan compartment	Filled
4	3	0	0	Rear right fan compartment	Filled
5	1	0	0	Host controller compartment	Filled
6	1	0	0	SCB slot	Filled
7	1	0	0	FPC slot 0	Empty
7	2	0	0	FPC slot 1	Empty
7	3	0	0	FPC slot 2	Filled
7	4	0	0	FPC slot 3	Filled
7	5	0	0	FPC slot 4	Empty
7	6	0	0	FPC slot 5	Filled
7	7	0	0	FPC slot 6	Empty

Container Index	L1	L2	L3	Description	State
7	8	0	0	FPC slot 7	Empty
8	1	1	0	PIC space @ 0/0/*	Empty
8	1	2	0	PIC space @ 0/1/*	Empty
8	1	3	0	PIC space @ 0/2/*	Empty
8	1	4	0	PIC space @ 0/3/*	Empty
8	2	1	0	PIC space @ 1/0/*	Empty
8	2	2	0	PIC space @ 1/1/*	Empty
8	2	3	0	PIC space @ 1/2/*	Empty
8	2	4	0	PIC space @ 1/3/*	Empty
8	3	1	0	PIC space @ 2/0/*	Filled
8	3	2	0	PIC space @ 2/1/*	Filled
8	3	3	0	PIC space @ 2/2/*	Filled
8	3	4	0	PIC space @ 2/3/*	Filled
8	4	1	0	PIC space @ 3/0/*	Filled
8	4	2	0	PIC space @ 3/1/*	Filled
8	4	3	0	PIC space @ 3/2/*	Filled
8	4	4	0	PIC space @ 3/3/*	Filled
8	5	1	0	PIC space @ 4/0/*	Empty
8	5	2	0	PIC space @ 4/1/*	Empty
8	5	3	0	PIC space @ 4/2/*	Empty
8	5	4	0	PIC space @ 4/3/*	Empty
8	6	1	0	PIC space @ 5/0/*	Filled
8	6	2	0	PIC space @ 5/1/*	Filled
8	6	3	0	PIC space @ 5/2/*	Filled
8	6	4	0	PIC space @ 5/3/*	Filled
8	7	1	0	PIC space @ 6/0/*	Empty
8	7	2	0	PIC space @ 6/1/*	Empty
8	7	3	0	PIC space @ 6/2/*	Empty
8	7	4	0	PIC space @ 6/3/*	Empty
8	8	1	0	PIC space @ 7/0/*	Empty
8	8	2	0	PIC space @ 7/1/*	Empty
8	8	3	0	PIC space @ 7/2/*	Empty
8	8	4	0	PIC space @ 7/3/*	Empty
9	1	0	0	Routing Engine compartment	Filled

Table 43 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T640 routing node.

Table 43: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Empty
2	2	0	0	PEM slot 1	Filled
4	1	1	0	Top left front fan slot	Filled
4	1	2	0	Top left middle fan slot	Filled
4	1	3	0	Top left rear fan slot	Filled
4	1	4	0	Top right front fan slot	Filled
4	1	5	0	Top right middle fan slot	Filled
4	1	6	0	Top right rear fan slot	Filled
4	2	1	0	Bottom left front fan slot	Filled
4	2	2	0	Bottom left middle fan slot	Filled
4	2	3	0	Bottom left rear fan slot	Filled
4	2	4	0	Bottom right front fan slot	Filled
4	2	5	0	Bottom right middle fan slot	Filled
4	2	6	0	Bottom right rear fan slot	Filled
4	3	1	0	Fourth blower from top slot	Filled
4	3	2	0	Bottom blower slot	Filled
4	3	3	0	Middle blower slot	Filled
4	3	4	0	Top blower slot	Filled
4	3	5	0	Second blower from top slot	Filled
7	3	2	0	FPC slot 0	Empty
7	3	3	0	FPC slot 0 top temp. sensor	Empty
7	3	4	0	FPC slot 0 bottom temp. sensor	Empty
7	3	5	0	FPC slot 1	Filled
7	3	6	0	FPC slot 1 top temp. sensor	Filled
7	1	0	0	FPC slot 1 bottom temp. sensor	Filled
7	1	1	0	FPC slot 2	Empty
7	1	2	0	FPC slot 2 top temp. sensor	Empty
7	2	0	0	FPC slot 2 bottom temp. sensor	Empty
7	2	1	0	FPC slot 3	Empty
7	2	2	0	FPC slot 3 top temp. sensor	Empty
7	3	0	0	FPC slot 3 bottom temp. sensor	Empty
7	3	1	0	FPC slot 4	Empty
7	3	2	0	FPC slot 4 top temp. sensor	Empty
7	4	0	0	FPC slot 4 bottom temp. sensor	Empty

Container Index	L1	L2	L3	Description	State
7	4	1	0	FPC slot 5	Filled
7	4	2	0	FPC slot 5 top temp. sensor	Filled
7	5	0	0	FPC slot 5 bottom temp. sensor	Filled
7	5	1	0	FPC slot 6	Empty
7	5	2	0	FPC slot 6 top temp. sensor	Empty
7	6	0	0	FPC slot 6 bottom temp. sensor	Empty
7	6	1	0	FPC slot 7	Filled
7	6	2	0	FPC slot 7 top temp. sensor	Filled
7	7	0	0	FPC slot 7 bottom temp. sensor	Filled
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	1	3	0	PIC slot @ 0/2/*	Empty
8	1	4	0	PIC slot @ 0/3/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Filled
8	2	2	0	PIC slot @ 1/1/*	Filled
8	2	3	0	PIC slot @ 1/2/*	Filled
8	2	4	0	PIC slot @ 1/3/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	3	3	0	PIC slot @ 2/2/*	Empty
8	3	4	0	PIC slot @ 2/3/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Empty
8	4	2	0	PIC slot @ 3/1/*	Empty
8	4	3	0	PIC slot @ 3/2/*	Empty
8	4	4	0	PIC slot @ 3/3/*	Empty
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	3	0	PIC slot @ 4/2/*	Empty
8	5	4	0	PIC slot @ 4/3/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Filled
8	6	2	0	PIC slot @ 5/1/*	Filled
8	6	3	0	PIC slot @ 5/2/*	Filled
8	6	4	0	PIC slot @ 5/3/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	7	3	0	PIC slot @ 6/2/*	Empty
8	7	4	0	PIC slot @ 6/3/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Filled

Container Index	L1	L2	L3	Description	State
8	8	2	0	PIC slot @ 7/1/*	Filled
8	8	3	0	PIC slot @ 7/2/*	Filled
8	8	4	0	PIC slot @ 7/3/*	Filled
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled
15	4	0	0	SIB slot 3	Filled
15	5	0	0	SIB slot 4	Filled

Table 44 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T320 router.

Table 44: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Filled
2	2	0	0	PEM slot 1	Empty
4	1	1	0	Top left front fan slot	Filled
4	1	2	0	Top left middle fan slot	Filled
4	1	3	0	Top left rear fan slot	Filled
4	1	4	0	Top right front fan slot	Filled
4	1	5	0	Top right middle fan slot	Filled
4	1	6	0	Top right rear fan slot	Filled
4	2	1	0	Bottom left front fan slot	Filled
4	2	2	0	Bottom left middle fan slot	Filled
4	2	3	0	Bottom left rear fan slot	Filled
4	2	4	0	Bottom right front fan slot	Filled
4	2	5	0	Bottom right middle fan slot	Filled
4	2	6	0	Bottom right rear fan slot	Filled
4	3	1	0	Rear tray top fan slot	Filled
4	3	2	0	Rear tray second fan slot	Filled
4	3	3	0	Rear tray middle fan slot	Filled
4	3	4	0	Rear tray fourth fan slot	Filled
4	3	5	0	Rear tray bottom fan slot	Filled
7	1	0	0	FPC slot 0	Empty
7	1	1	0	FPC slot top temp. sensor	Empty
7	1	2	0	FPC slot 0 bottom temp. sensor	Empty
7	2	0	0	FPC slot 1	Empty
7	2	1	0	FPC slot 1 top temp. sensor	Empty
7	2	2	0	FPC slot 1 bottom temp. sensor	Empty
7	3	0	0	FPC slot 2	Empty
7	3	1	0	FPC slot 2 top temp. sensor	Empty
7	3	2	0	FPC slot 2 bottom temp. sensor	Empty
7	4	0	0	FPC slot 3	Filled
7	4	1	0	FPC slot 3 top temp. sensor	Filled
7	4	2	0	FPC slot 3 bottom temp. sensor	Filled
7	5	1	0	FPC slot 4	Empty
7	5	2	0	FPC slot 4 top temp. sensor	Empty
7	5	0	0	FPC slot 4 bottom temp. sensor	Empty

Container Index	L1	L2	L3	Description	State
7	6	1	0	FPC slot 5	Empty
7	6	2	0	FPC slot 5 top temp. sensor	Empty
7	6	0	0	FPC slot 5 bottom temp. sensor	Empty
7	7	1	0	FPC slot 6	Empty
7	7	2	0	FPC slot 6 top temp. sensor	Empty
7	7	0	0	FPC slot 6 bottom temp. sensor	Empty
7	8	1	0	FPC slot 7	Empty
7	8	2	0	FPC slot 7 top temp. sensor	Empty
7	8	0	0	FPC slot 7 bottom temp. sensor	Empty
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Empty
8	2	2	0	PIC slot @ 1/1/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Filled
8	4	2	0	PIC slot @ 3/1/*	Filled
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Empty
8	6	2	0	PIC slot @ 5/1/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Empty
8	8	2	0	PIC slot @ 7/1/*	Empty
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled

jnxOperatingTable

The object identifier for jnxOperatingTable object is {jnxBoxAnatomy 13}. This object reports the operating status of various components such as CPU, buffers, and memory.

Juniper Networks routers implement packet forwarding and routing functions with two separate components, the Packet Forwarding Engine and the Routing Engine, to ensure stability. The clean separation of these two functions permits superior forwarding performance and a highly reliable operating system. Therefore, it is not necessary to monitor CPU, memory, and buffer utilization, as is the case with traditional, monolithic code base routers. The Routing Engine has its own CPU, memory, and buffers—separate from those of the Packet Forwarding Engine. The ASIC-based Packet Forwarding Engine forwards packets on all interfaces at wire speed, eliminating the need to monitor packet buffers being exhausted. As a result, CPU utilization under 2 percent is normal.

Entries in the jnxOperatingTable are represented by the jnxOperatingEntry object, whose object identifier is {jnxOperatingTable 1}.

The jnxOperating Table describes the status of specific objects, which are described as follows:

- jnxOperatingContents—The associated jnxContentsIndex in the jnxContentsTable, whose object identifier is {jnxOperatingEntry 1}.
- jnxOperatingL1Index—The level-one index of the container housing the entry, whose object identifier is {jnxOperatingEntry 2}.
- jnxOperatingL2Index—The level-two index of the container housing the entry, whose object identifier is {jnxOperatingEntry 3}.
- jnxOperatingL3Index—The level-three index of the container housing the entry, whose object identifier is {jnxOperatingEntry 4}.
- jnxOperatingDescr—The name or detailed description of the entry, whose object identifier is {jnxOperatingEntry 5}.
- jnxOperatingState—The operating state of the entry, whose object identifier is {jnxOperatingEntry 6}. The state can be any of the following:
  - Unknown(1)—State of the component is unknown or unavailable
  - Running(2)—Up and running as an active primary
  - Ready(3)—Ready to run; not running yet
  - Reset(4)—Held in reset; not ready yet
  - RunningAtFullSpeed(5)—Valid for fans only
  - Down(6)—Power supply is down or off
  - Standby(7)—Running as a standby backup

- jnxOperatingTemp—The entry's temperature, in degrees Celsius (°C), whose object identifier is {jnxOperatingEntry 7}.

- jnxOperatingCPU—The CPU utilization percentage of the entry, whose object identifier is {jnxOperatingEntry 8}. It is valid for the Control Board, the FPC, and the Routing Engine. It is a 5-second rolling weighted average calculated every second for each of the CPUs. The value is sent to the Routing Engine every 10 seconds. The value for the Routing Engine is an average of samples taken every 30 seconds over a 5-minute period. jnxOperatingCPU.9.1.0.0. is for the Routing Engine CPU. The Routing Engine is the only object of interest; the rest are most likely zero because CPUs on those cards are only used for management purposes.

- jnxOperatingISR—The CPU utilization percentage of the entry in relation to the interrupt service routing (ISR), whose object identifier is {jnxOperatingEntry 9}.

- jnxOperatingDRAMSize—The DRAM size of the entry, in bytes, whose object identifier is {jnxOperatingEntry 10}. It is valid for the FPC, Routing Engine, and Control Board.

- jnxOperatingBuffer—The buffer pool utilization of the entry (a percentage), whose object identifier is {jnxOperatingEntry 11}. It is valid for the FPC and Control Board as a percentage of utilization. Buffers are normally fixed-length memory preallocated for read/write, input/output, or reception/transmission. A measurement against these buffers gives some indication of how busy the system is. The larger the percentage utilization, the busier the system. In terms of absolute numbers, the bigger the buffer size, the better the system can handle bursty traffic patterns.

- jnxOperatingHeap—The heap utilization of the entry, whose object identifier is {jnxOperatingEntry 12}.

- jnxOperatingUpTime—The time interval, in 10-millisecond periods, that the entry has been up and running, whose object identifier is {jnxOperatingEntry 13}.

- jnxOperatingLastRestart—The value of sysUpTime when the entry was last restarted, whose object identifier is {jnxOperatingEntry 14}.

- jnxOperatingMemory—The entry's installed memory size, in megabytes (MB), whose object identifier is {jnxOperatingEntry 15}.

- jnxOperatingStateOrdered—The operating state of the entry, whose object identifier is {jnxOperatingEntry 16}. The state can be any of the following

  - Running(1)—Up and running as an active primary

  - Standby(2)—Running as a standby backup

  - Ready(3)—Ready to run; not running yet

  - RunningAtFullSpeed(4)—Valid for fans only

  - Reset(5)—Held in reset; not ready yet

- Down(6)—Power supply is down or off
- Unknown(7)—State of the component is unknown or unavailable

Table 45 through Table 47 provide examples of jnxOperatingEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxOperatingEntry objects:

- Contents index—jnxOperatingContents
- L1—jnxOperatingL1Index
- L2—jnxOperatingL2Index
- L3—jnxOperatingL3Index
- Description—jnxOperatingDescr
- State—jnxOperatingState
- Temp—jnxOperatingTemp
- CPU—jnxOperatingCPU
- ISR—jnxOperatingISR
- DRAM—jnxOperatingDRAMSize
- Buffer—jnxOperatingBuffer
- Heap—jnxOperatingHeap
- UpTime—jnxOperatingUpTime
- Last Restart—jnxOperatingLastRestart
- Memory—jnxOperatingMemory

Table 45 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of an M20 router.

Table 45: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
1	1	0	0	Midplane	Running	26	0	0	0	0	0	0	0:0:00:00.0	0
2	1	0	0	Power supply A	Running	28	0	0	0	0	0	0	0:0:00:00.0	0
2	2	0	0	Power supply B	Running	29	0	0	0	0	0	0	0:0:00:00.0	0
4	1	0	0	Front top fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	2	0	0	Front middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	3	0	0	Front bottom fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	4	0	0	Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
6	1	0	0	SSB 0	Running	30	0	0	67108864	6	0	67038195	0:0:00:35.41	64
7	1	0	0	FPC @ 0/*/*	Running	31	0	0	8388608	3	0	67035034	0:0:01:06.91	8
7	2	0	0	FPC @ 1/*/*	Running	33	0	0	8388608	4	0	67034422	0:0:01:13.04	8
7	3	0	0	FPC @ 2/*/*	Running	31	0	0	8388608	3	0	67033809	0:0:01:19.18	8
9	1	0	0	Routing Engine 0	Running	29	4	0	802738176	0	0	67046146	0:0:00:00.00	765

To verify the size of the memory, use the show chassis fpc, show chassis routing-engine, and show chassis ssb commands. For more information on the output of these commands, see the *JUNOS System Basics and Services Command Reference*.

Table 46 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of a T640 routing node.

Table 46: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Rstrt	Memory
1	1	0	0	Midplane	Running	0								
2	2	0	0	PEM 1	Running	29								
4	1	1	0	Top left front fan	Running	0								
4	1	2	0	Top left middle fan	Running	0								
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	6	0	Top right rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom right rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	2	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	3	0	Middle blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	4	0	Top blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	5	0	Second blower from top	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	Running	0	1	0	512	41	3	138367	0:18:56:48.81	512
7	2	1	0	FPC @ 1/0/* top temp. sensor	Running	35	0	0	0	0	0	0	0:18:56:48.81	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Rstrt	Memory
7	2	2	0	FPC @ 1/1/* bottom temp. sensor	Running	32	0	0	0	0	0	0	0:18:56 :48.81	0
7	6	0	0	FPC @ 5/*/*	Running	0	3	0	256	41	14	136976	0:18:57 :02.71	256
7	6	1	0	FPC @ 5/0/* top temp. sensor	Running	44	0	0	0	0	0	0	0:18:57 :02.71	0
7	6	2	0	FPC @ 5/1/* bottom temp. sensor	Running	33	0	0	0	0	0	0	0:18:57 :02.71	0
7	8	0	0	FPC @ 7/*/*	Running	0	2	0	256	41	7	137963	0:18:56 :52.85	256
7	8	1	0	FPC @ 7/0/* top temp. sensor	Running	38	0	0	0	0	0	0	0:18:56 :52.85	0
7	8	2	0	FPC @ 7/1/* bottom temp. sensor	Running	33	0	0	0	0	0	0	0:18:56 :52.85	0
9	1	0	0	Host 0	Running	35	0	0	2048	0	0	6963005	0:19:20 :30.07	2048
9	2	0	0	Host 1	Standby	32	2	0	2048	0	0	24401100	2:19:46 :51.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	0:0:00: 00.00	0
11	1	0	0	SCG 0	Running	36	0	0	0	0	0	0	0:0:00: 00.00	0
11	2	0	0	SCG 1	Standby	35	0	0	0	0	0	0	0:0:00: 00.00	0
12	1	0	0	CB 0	Running	36	0	0	0	0	0	0	0:0:00: 00.00	0
12	2	0	0	CB 1	Standby	39	0	0	0	0	0	0	0:0:00: 00.00	0
14	1	0	0	SPMB 0	Running	36	1	0	128	40	0	142576	0:18:56 :06.72	128
14	2	0	0	SPMB 1	Standby	39	0	0	128	40	0	142447	0:18:56 :08.01	128
15	1	0	0	SIB 0	Unknown	40	0	0	0	0	0	0	0:0:00: 00.00	0
15	2	0	0	SIB 1	Unknown	39	0	0	0	0	0	0	0:0:00: 00.00	0
15	3	0	0	SIB 2	Unknown	39	0	0	0	0	0	0	0:0:00: 00.00	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Rstrt	Memory
15	4	0	0	SIB 3	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0
15	5	0	0	SIB 4	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0

Table 47 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of a T320 router.

Table 47: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
2	1	0	0	PEM 0	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	1	0	Top left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	2	0	Top left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Top right rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	1	0	Bottom left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	2	0	Bottom left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Bottom right rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	1	0	Rear tray top fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	2	0	Rear tray second fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	3	0	Rear tray middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	4	0	Rear tray fourth fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	5	0	Rear tray bottom fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	Running	0	1	0	256	41	7	6568428	(26190949) |3 days, 0:45:09.49	256
7	4	1	0	FPC @ 3/0/* top temp. sensor	Running	41	0	0	0	0	0	0	26190949) 3 days, 0:45:09.49	0
7	4	2	0	FPC @ 3/1/* bottom temp. sensor	Running	37	0	0	0	0	0	0	(26190949) 3 days, 0:45:09.49	0
9	1	0	0	Host 0	Running	34	1	0	2048	0	0	32763001	(32763004) 3 days, 19:00:30.04	2048
9	2	0	0	Host 1	Standby	32	1	0	2048	0	0	110271900	(110271900) 12 days, 18:18:39.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	(0) 0:00:00.00	0
11	1	0	0	SCG 0	Running	33	0	0	0	0	0	0	(0) 0:00:00.00	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
11	2	0	0	SCG 1	Standby	31	0	0	0	0	0	0	(0) 0:00:00.00	0
12	1	0	0	CB 0	Running	37	0	0	0	0	0	0	(0) 0:00:00.00	0
12	2	0	0	CB 1	Standby	34	0	0	0	0	0	0	(0) 0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	0	0	128	40	0	6572381	(26186997) 3 days, 0:44:29.97	128
14	2	0	0	SPMB 1	Standby	36	1	0	128	40	0	6572465	(26186913) 3 days, 0:44:29.13	128
15	1	0	0	SIB 0	Standby	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	2	0	0	SIB 1	Running	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	3	0	0	SIB 2	Running	38	0	0	0	0	0	0	(0) 0:00:00.00	0

jnxRedundancyTable

The object identifier for the jnxRedundancyTable is {jnxBoxAnatomy 14}. This object shows the internal configuration settings for the redundant subsystems or components in the chassis.

Entries within the jnxRedundancyTable are represented by the jnxRedundancyEntry object, whose object identifier is {jnxRedundancyEntry 1}. This jnxRedundancyEntry contains the following objects, which describe the internal configuration settings for the redundant subsystems or components in the chassis:

- jnxRedundancyContentsIndex—The index value of an entry in jnxRedundancyEntry, whose object identifier is {jnxContainersEntry 1}.
- jnxRedundancyL1Index—The level-one index associated with the redundant component, whose object identifier is {jnxContainersEntry 2}.
- jnxRedundancyL2Index—The level-two index associated with the redundant component, whose object identifier is {jnxContainersEntry 3}.
- jnxRedundancyL3Index—The level-three index associated with the redundant component, whose object identifier is {jnxContainersEntry 4}.
- jnxRedundancyDescr—The description of the redundant component, whose object identifier is {jnxContainersEntry 5}.
- jnxRedundancyConfig—The election priority of redundancy configuration, whose object identifier is {jnxContainersEntry 6}.
- jnxRedundancyState—The current running state of the redundant component, whose object identifier is {jnxContainersEntry 7}.
- jnxRedundancySwitchoverCount—The total number of switchovers, defined as a change in the jnxRedundancyState from master to backup or vice versa, as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is {jnxContainersEntry 8}.
- jnxRedundancySwitchoverTime—The value of sysUpTime when the jnxRedundancyState was last switched over from master to backup or vice versa, whose object identifier is {jnxContainersEntry 9}.
- jnxRedundancySwitchoverReason—The reason for the last switchover to the redundant component, whose object identifier is {jnxContainersEntry 10}.
- jnxKeepaliveHeartbeat—The period of sending keepalive messages between the master and the backup subsystem, which is a system-wide preset value in seconds used by internal mastership resolution, whose object identifier is {jnxContainersEntry 11}.
- jnxRedundancyKeepaliveTimeout—The timeout period in seconds used by the watchdog timer before it initiates a switchover to the backup subsystem, whose object identifier is {jnxContainersEntry 12}.

- jnxRedundancyKeepaliveElapsed—The elapsed time since the redundant component received the last keepalive message from the outer subsystems, whose object identifier is {jnxContainersEntry 13}.

- jnxRedundancyKeepaliveLoss—The total number of keepalive messages lost between the master and the backup subsystems as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is {jnxContainersEntry 14}.

Table 48 through Table 50 provide examples of jnxRedundancyEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxOperatingTable objects:

- Contents index—jnxRedundancyContentsIndex
- L1—jnxRedundancyL1Index
- L2—jnxRedundancyL2Index
- L3—jnxRedundancyL3Index
- Description—jnxRedundancyDescr
- Config—jnxRedundancyConfig
- State—jnxRedundancyState
- Count—jnxRedundancySwitchoverCount
- Time—jnxRedundancySwitchoverTime
- Reason—jnxRedundancySwitchoverReason
- Heartbeat—jnxKeepaliveHeartbeat
- Timeout—jnxRedundancyKeepaliveTimeout
- Elapsed—jnxRedundancyKeepaliveElapsed
- Loss—jnxRedundancyKeepaliveLoss

Table 48 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of an M20 router.

Table 48: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart-beat	Timeout	Elapsed	Loss
6	1	0	0	SSB 0 Internet Processor II	Master	Master	0	3383	Never switched	0	0	0	0
6	2	0	0	SSB 1	Disabled	Disabled	0	0	Never switched	0	0	0	0
9	1	0	0	Routing Engine 0	Master	Master	1	421	User switched	3	300	1	0
9	2	0	0	Routing Engine 1	Backup	Backup	0	0	Other	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M20 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
 Slot 0:
  Current state     Master
  Election priority Master (default)
  Temperature       26 degrees C / 78 degrees F
  DRAM              768 Mbytes
  CPU utilization:
   User             2 percent
   Background       0 percent
   Kernel           0 percent
   Interrupt         0 percent
   Idle             98 percent
  Model             teknor
  Serial ID         32000004f8ff1201
  Start time        2002-01-29 12:30:42 PST
  Uptime             21 hours, 17 minutes, 14 seconds
  Load averages:    1 minute  5 minute 15 minute
    0.03    0.02    0.00
Routing Engine status:
 Slot 1:
  Current state     Backup
  Election priority Backup (default)
  DRAM              805306368 Mbytes
  CPU utilization:
   User             0 percent
   Background       0 percent
   Kernel           1 percent
   Interrupt         0 percent
   Idle             99 percent
  Model             teknor
  Serial ID         100000078c10df01
  Start time        2002-01-24 16:47:39 PST
  Uptime             5 days, 17 hours, 14 seconds
```

To verify SSB status, use the show chassis ssb command. Sample command output from an M20 router is listed below.

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
 State                       Master
 Temperature                 24 degrees C / 75 degrees F
 CPU utilization             2 percent
 Interrupt utilization       0 percent
 Heap utilization            16 percent
 Buffer utilization          43 percent
 Total CPU DRAM              64 Mbytes
 Internet Processor II       Version 1, Foundry IBM, Part number 9
 Start time:                 2002-01-29 12:32:24 PST
 Uptime:                     21 hours, 30 minutes, 53 seconds
Slot 1 information:
 State                       Backup
```

Table 49 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of a T640 routing node.

Table 49: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Routing Node

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart-beat	Timeout	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	3	0:18:55:49.42	User switched	20	300	1	0
9	2	0	0	Host 1	Backup	Backup	0	0:0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Unknown	Backup	1	0:0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	4	0	0	SIB 3	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	5	0	0	SIB 4	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state       Master
    Election priority   Master (default)
    Temperature         35 degrees C / 95 degrees F
    DRAM                2048 MB
    CPU utilization:
      User              1 percent
      Background        0 percent
      Kernel            5 percent
      Interrupt         0 percent
      Idle              94 percent
    Model               unknown
    Start time     2002-03-31 14:26:49 PST
    Uptime         19 hours, 22 minutes, 13 seconds
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority              Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
    Start time     2002-03-29 14:00:18 PST
    Uptime         2 days, 19 hours, 48 minutes, 32 seconds
```

Table 50 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of a T320 router.

Table 50: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heartbeat	Timeout	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	6	(26185188) 3 days, 0:44:11.88	User switched	20	300	1	0
9	2	0	0	Host 1	Backup	Backup	0	(0) 0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Backup	Backup	1	(0) 0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from a T320 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state  Master
    Election priority             Master (default)
    Temperature 34 degrees C / 93 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     1 percent
      Interrupt  0 percent
      Idle      98 percent
    Model          RE-3.0
    Start time     2002-04-05 14:43:16 PST
    Uptime         17 days, 23 hours, 3 minutes, 47
seconds
```

```
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority              Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
    Start time     2002-03-27 15:25:07 PST
    Uptime         26 days, 22 hours, 21 minutes, 44 seconds
```

jnxFruTable

The object identifier for the jnxFruTable is {jnxBoxAnatomy 15}. This object shows the status of field-replaceable units (FRUs) in the chassis.

Entries within the jnxFruTable are represented by the jnxFruEntry object, whose object identifier is {jnxFruEntry 1}. This jnxFruEntry object contains the following objects, which describe the FRUs in the chassis:

- jnxFruContentsIndex—The index value of an entry in jnxFruEntry, whose object identifier is {jnxFruEntry 1}.
- jnxFruL1Index—The level-one index associated with the FRU, whose object identifier is {jnxFruEntry 2}.
- jnxFruL2Index—The level-two index associated with the FRU, whose object identifier is {jnxFruEntry 3}.
- jnxFruL3Index—The level-three index associated with the FRU, whose object identifier is {jnxFruEntry 4}.
- jnxFruName—The name or detailed description of the FRU, whose object identifier is {jnxFruEntry 5}.
- jnxFruType—The FRU type, whose object identifier is {jnxFruEntry 6}. The FRU type can be any of the following:
  - other(1)
  - clockGenerator(2)
  - flexiblePicConcentrator(3)
  - switchingAndForwardingModule(4)
  - controlBoard(5)
  - routingEngine(6)
  - powerEntryModule(7)

- frontPanelModule(8)
- switchInterfaceBoard(9)
- processorMezzanineBoardForSIB(10)
- portInterfaceCard(11)
- craftInterfacePanel(12)
- fan(13)

- jnxFruSlot—The slot number of the FRU, whose object identifier is {jnxFruEntry 7}. This is equivalent to jnxFruL1Index. The slot number is zero if unavailable or inapplicable.
- jnxFruState—The current state of the FRU, whose object identifier is {jnxFruEntry 8}. The FRU state can be any of the following:
  - unknown(1)
  - empty(2)
  - present(3)
  - ready(4)
  - announceOnline(5)
  - online(6)
  - announceOffline(7)
  - offline(8)
  - diagnostic(9)
  - standby(10)
- jnxFruTemp—The temperature of the FRU, in degrees Celsius, whose object identifier is {jnxFruEntry 9}. The value is zero if unavailable or inapplicable.
- jnxFruOfflineReason—The reason the FRU is offline, whose object identifier is {jnxFruEntry 10}. The reason can be any of the following:
  - unknown(1)—Unknown or other
  - none(2)—None
  - error(3)—Error
  - noPower(4)—No power
  - configPowerOff(5)—Configured to power off
  - configHoldInReset(6)—Configured to hold in reset
  - cliCommand(7)—Brought offline by CLI command

  - buttonPress(8)—Brought offline by button press
  - cliRestart(9)—Restarted by CLI command
  - overtempShutdown(10)—Overtemperature shutdown
  - masterClockDown(11)—Master clock down
  - singleSfmModeChange(12)—Single SFM mode change
  - packetSchedulingModeChange(13)—Packet scheduling mode change
  - physicalRemoval(14)—Physical removal
  - unresponsiveRestart(15)—Restarting unresponsive board
  - sonetClockAbsent(16)—SONET out clock absent

- jnxFruLastPowerOff—The value of sysUpTime when this subject was last powered off, whose object identifier is {jnxFruEntry 11}. The value is zero if unavailable or inapplicable.
- jnxFruLastPowerOn—The value of sysUpTime when this subject was last powered on, whose object identifier is {jnxFruEntry 12}. The value is zero if unavailable or inapplicable.
- jnxFruPowerUpTime—The time interval in 10-millisecond periods that this subject has been up and running since the last power-on time, whose object identifier is {jnxFruEntry 13}. The value is zero if unavailable or inapplicable.

Table 51 through Table 56 provide examples of jnxFruEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxFruEntry objects:

- Contents Index—jnxFruContentsIndex
- L1—jnxFruL1Index
- L2—jnxFruL2Index
- L3—jnxFruL3Index
- Name—jnxFruName
- Type—jnxFruType
- Slot—jnxFruSlot
- State—jnxFruState
- Temp—jnxFruTemp
- Offline—jnxFruOffline
- PowerOff—jnxFruPowerOff

- PowerOn—jnxFruPowerOn
- Uptime—jnxFruPowerUpTime

Table 51 provides an example of jnxFruContent objects in the jnxFruTable for an M10 router.

Table 51: jnxFruContents Objects in the jnxFruTable of an M10 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntryModule	1	online	0	none	0:0:00:00.00	0:0:11:08.73	26431910
2	2	0	0	Power supply B	powerEntryModule	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	1	0	Left fan 1	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Left fan 2	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Left fan 3	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Left fan 4	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	FEB Internet Processor II	controlBoard	1	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x Monitor @ 0/1/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 1x OC-12 ATM, MM @ 0/2/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 4x T3 @ 0/3/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 4x OC-3 SONET, SMIR @ 1/0/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 2x OC-3 ATM, MM @ 1/3/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M10 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                58974             M10
Midplane         REV 03   710-001950   HB1590
Power Supply A   Rev 03   740-002498   LK33505           DC
Display          REV 04   710-001995   HE8442
Routing Engine   REV 01   740-003239   9001025728        RE-2.0
FEB              REV 12   710-001948   HA4221            Internet Processor II
FPC 0
  PIC 1          REV 01   750-004188   AR2912            1x Monitor
  PIC 2          REV 04   750-001551   AN7869            1x OC-12 ATM, MM
  PIC 3          REV 02   750-002485   AN2803            4x T3
FPC 1
  PIC 0          REV 03   750-002970   HF2293            4x OC-3 SONET, SMIR
  PIC 1          REV 03   750-002971   HA8094            4x OC-3 SONET, MM
  PIC 2          REV 03   750-002977   HD9352            2x OC-3 ATM, MM
  PIC 3          REV 03   750-002977   HD9393            2x OC-3 ATM, MM
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M10 router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            24      3          1        64        44         17
  1  Online            24      3          1        64        44         17
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M10 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                26 degrees C / 78 degrees F
    DRAM                      768 MB
    Memory utilization          9 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    0 percent
      Interrupt                 0 percent
      Idle                    100 percent
```

```
Model                          RE-2.0
Serial ID                      b7000007c81ce801
Start time                     2002-06-21 09:33:45 PDT
Uptime                         3 days, 1 hour, 23 minutes, 27 seconds
Load averages:                 1 minute   5 minute  15 minute
                                   0.07       0.03       0.01
```

To verify FEB status, use the show chassis feb command. Sample command output from an M10 router is listed below.

```
user@host> show chassis feb
FEB status:
  Temperature                 24 degrees C / 75 degrees F
  CPU utilization              3 percent
  Interrupt utilization        1 percent
  Heap utilization            17 percent
  Buffer utilization          44 percent
  Total CPU DRAM              64 MB
  Internet Processor II     Version 1, Foundry IBM, Part number 9
  Start time:               2002-06-21 09:45:46 PDT
  Uptime:                   3 days, 1 hour, 11 minutes, 33 seconds
```

Table 52 provides an example of jnxFruContent objects in the jnxFruTable for an M20 router.

Table 52: JnxFruContents Objects in the jnxFruTable of an M20 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntryModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	Power supply B	powerEntryModule	2	online	25	none	0:0:00:00.00	0:0:00:43.45	24993357
4	1	0	0	Rear fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Front upper fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Front middle fan	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Front bottom fan	fan	4	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SSB 0	controlBoard	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	2	0	0	SSB 1 Internet Processor I	controlBoard	2	online	29	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	27	none	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	27	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	offline	28	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x Tunnel @ 1/0/*	portInterfaceCard	2	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x T3 @ 1/1/*	portInterfaceCard	2	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x G/E, 1000 BASE-SX @ 1/3/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 2x OC-3 SONET, SMIR @ 3/2/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	online	25	none	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	2	online	24	none	0:0:00:00.00	0:0:00:00.00	0
10	1	0	0	Front panel display	frontPanelModule	1	online	0	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M20 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                20200             M20
Backplane        REV 07   710-001517   AB5911
Power Supply B   Rev 02   7            000240            AC
Display          REV 04   710-001519   AD1903
Routing Engine 0 REV01    740          umeshk            RE-2.0
Routing Engine 1                       270000078ba48501  RE-2.0
SSB slot 0       N/A      N/A          N/A               backup
SSB slot 1       REV 04   710-001411   AD0281            Internet Processor I
FPC 1            REV 01   710-001292   AC9230
  PIC 0          REV 01   750-001323   AA2812            1x Tunnel
  PIC 1          REV 01   750-002963   AK8586            4x T3
  PIC 2          REV 03   750-000612   AM8116            2x OC-3 ATM, MM
  PIC 3          REV 08   750-001072   AB9884            1x G/E, 1000 BASE-SX
FPC 3            REV 01   710-001197   AA8661
  PIC 2          REV 01   750-003748   HE9734            2x OC-3 SONET, SMIR

user@host> show chassis environment
Class Item                 Status     Measurement
Power Power Supply A       Absent
      Power Supply B       OK         25 degrees C / 77 degrees F
Temp  FPC 1                OK         27 degrees C / 80 degrees F
      FPC 3                OK         28 degrees C / 82 degrees F
      SSB 1                OK         29 degrees C / 84 degrees F
      Backplane            OK         23 degrees C / 73 degrees F
      Routing Engine 0     OK         25 degrees C / 77 degrees F
      Routing Engine 1     OK         24 degrees C / 75 degrees F
Fans  Rear Fan             OK         Spinning at normal speed
      Front Upper Fan      OK         Spinning at normal speed
      Front Middle Fan     OK         Spinning at normal speed
      Front Bottom Fan     OK         Spinning at normal speed
Misc  Craft Interface      OK

user@host> show chassis fpc
                    Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty             0      0          0         0         0          0
  1  Online           27      8          7         8         9         14
  2  Empty             0      0          0         0         0          0
  3  Online           28      0          0         8         8         14
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M10 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 25 degrees C / 77 degrees F
    DRAM                       768 MB
    Memory utilization           8 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     1 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                        RE-2.0
    Serial ID                    ba0000061779d601
    Start time                   2002-06-21 15:37:36 PDT
    Uptime                       2 days, 21 hours, 27 minutes, 25 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                 24 degrees C / 75 degrees F
    DRAM                       768 MB
    Memory utilization           9 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     0 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                        RE-2.0
    Serial ID                    270000078ba48501
    Start time                   2002-06-17 14:30:21 PDT
    Uptime                       6 days, 22 hours, 34 minutes, 28 seconds
```

To verify SSB status, use the show chassis SSB command. Sample command output from an M10 router is listed below.

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
  State                        Backup
Slot 1 information:
  State                        Master
  Temperature                 29 degrees C / 84 degrees F
  CPU utilization              1 percent
  Interrupt utilization        0 percent
  Heap utilization             8 percent
  Buffer utilization          43 percent
  Total CPU DRAM              64 MB
  Internet Processor I         Version 1, Foundry IBM, Part number 3
  Start time:                  2002-06-21 15:38:53 PDT
  Uptime:                      2 days, 21 hours, 26 minutes, 26 seconds
```

Table 53 provides an example of jnxFruContent objects in the jnxFruTable for an M160 router.

Table 53: jnxFruContents Objects in the jnxFruTable of an M160 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	online	0	none	0:00:00.00	0:00:12.83	6906955
2	2	0	0	PEM 1	powerEntryModule	2	online	0	none	0:00:00.00	0:00:12.83	6906955
4	1	0	0	Front top blower	fan	1	present	0	none	0:00:00.00	0:00:00.00	0
4	2	1	0	Fan tray front left	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	2	0	Fan tray front right	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	3	0	Fan tray rear left	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	4	0	Fan tray rear right	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	3	0	0	Rear top blower	fan	3	present	0	none	0:00:00.00	0:00:00.00	0
4	4	0	0	Rear bottom blower	fan	4	present	0	none	0:00:00.00	0:00:00.00	0
6	1	1	0	SFM 0 SPP	switchingAnd-ForwardingMode	1	online	35	none	0:00:03.13	0:00:00.00	0
6	1	2	0	SFM 0 SPR Internet Processor II	switchingAnd-ForwardingMode	1	online	35	none	0:00:03.13	0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAnd-ForwardingMode	2	empty	0	none	0:00:00.00	0:00:00.00	0
6	2	2	0	SFM 1 SPR	switchingAnd-ForwardingMode	2	empty	0	none	0:00:00.00	0:00:00.00	0
6	3	1	0	SFM 2 SPP	switchingAnd-ForwardingMode	3	online	44	none	0:00:03.20	0:00:00.00	0
6	3	2	0	SFM 2 SPR Internet Processor II	switchingAnd-ForwardingMode	3	online	44	none	0:00:03.20	0:00:00.00	0
6	4	1	0	SFM 3 SPP	switchingAnd-ForwardingMode	4	offline	0	config-ured to power off	0:00:03.22	0:00:00.00	0
6	4	2	0	SFM 3 SPR	switchingAnd-ForwardingMode	4	offline	0	config-ured to power off	0:00:03.22	0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	offline	0	error	0:13:08.12	0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	online	30	none	0:00:02.32	0:00:00.00	0
7	4	0	0	FPC: 1x OC-192 SM LR @ 3/*/*	flexiblePicConcentrator	4	online	41	none	0:00:02.34	0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	empty	0	none	0:00:00.00	0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	offline	0	configured to power off	0:00:02.37	0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePicConcentrator	7	empty	0	none	0:00:00.00	0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	online	41	none	0:00:03.11	0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	config-ured to power off	0:00:02.28	0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	config-ured to power off	0:00:02.28	0:00:00.00	0
8	4	1	0	PIC: 1x OC-192 SM LR @ 3/0/*	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	2	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	3	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	4	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard (11)	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard (11)	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	8	1	0	PIC: 1x OC-12 SONET, SMIR @ 7/0/*	portInterfaceCard	8	offline	0	config-ured to power off	0:00:02.37	0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	2	0	PIC: 4x E3 @ 7/1/*	portInterfaceCard	8	offline	0	config-ured to power off	0:00:02.37	0:00:00.00	0
8	8	3	0	PIC: 1x OC-12 SONET, MM @ 7/2/* jnxFruName	portInterfaceCard	8	offline	0	config-ured to power off	0:00:02.37	0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	offline	0	config-ured to power off	0:00:02.37	0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	online	31	none	0:00:00.00	0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	2	present	0	none	0:00:00.00	0:00:00.00	0
10	1	1	0	FPM CMB	frontPanelModule	1	online	28	none	0:00:00.00	0:00:00.00	0
10	1	2	0	FPM Display	frontPanelModule	1	online	28	none	0:00:00.00	0:00:00.00	0
11	1	0	0	PCG 0	clockGenerator	1	online	40	none	0:00:00.00	0:00:00.00	0
11	2	0	0	PCG 1	clockGenerator	2	online	46	none	0:00:00.00	0:00:00.00	0
12	1	0	0	MCS 0	controlBoard	1	online	47	none	0:00:00.00	0:00:00.00	0
12	2	0	0	MCS 1	controlBoard	2	empty	0	none	0:00:00.00	0:00:00.00	0
13	1	0	0	CIP	craftInterface-Panel	1	present	0	none	0:00:00.00	0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M160 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                47                M160
Midplane         REV 02   710-001245   AB4113
FPM CMB          REV 01   710-001642   AA9721
FPM Display      REV 01   710-001647   AA2995
CIP              REV 02   710-001593   AA9886
PEM 0            Rev 01   740-001243   KJ35782           DC
PEM 1            Rev 01   740-001243   kj35756           DC
PCG 0            REV 01   710-001568   AA9796
PCG 1            REV 01   710-001568   AA9895
Routing Engine 0 REV01    740-003239   AARCHOO           RE-2.0
Routing Engine 1
MCS 0            REV 03   710-001226   AA9779
SFM 0 SPP        REV 07   710-001228   AE5504
SFM 0 SPR        REV 03   710-002189   AE4707            Internet Processor II
SFM 2 SPP        REV 06   710-001228   AB3133
SFM 2 SPR        REV 01   710-002189   AB2941            Internet Processor II
SFM 3 SPP        REV 07   710-001228   AV3167
SFM 3 SPR        REV 04   710-002189   AV3439            Internet Processor II
FPC 0            REV 02   710-001611   AA9518            FPC Type 2
  CPU            REV 02   710-001217   AA9572
FPC 1            REV 03   710-001255   AA9812            FPC Type 1
  CPU
FPC 2            REV 02   710-001611   AA9527            FPC Type 2
  CPU            REV 02   710-001217   AA9592
FPC 3            REV 01   710-003061   HB2029            FPC Type OC192
  CPU            REV 05   710-001217   AF5950
  PIC 0          REV 01   750-003063   HB2029            1x OC-192 SM LR
FPC 5            REV 01   710-001255   AA2914            FPC Type 1
  CPU            REV 02   710-001217   AA2893
FPC 7            REV 03   710-001255   AA9809            FPC Type 1
  CPU            REV 02   710-001217   AA9573
  PIC 0          REV 04   750-000613   AA0374            1x OC-12 SONET, SMIR
  PIC 1          REV 02   750-E3-PIC   AC1903            4x E3
  PIC 2          REV 02   750-001020   AA8944            1x OC-12 SONET, MM
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M160 router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Announce offline   0      0          0         0         0          0
  1  Present            0      0          0         0         0          0
  2  Online            32      4          0        32         1         39
  3  Online            44      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Offline         --- Chassis connection dropped ---
  6  Empty              0      0          0         0         0          0
  7  Online            42      4          0        32         1         40
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M160 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                       768 MB
    Memory utilization          10 percent
    CPU utilization:
      User                       1 percent
      Background                 0 percent
      Kernel                    10 percent
      Interrupt                  3 percent
      Idle                      87 percent
    Model                          RE-2.0
    Serial ID                      0c000004f8d26401
    Start time                     2002-06-14 14:39:03 PDT
    Uptime                         11 minutes, 46 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.18       0.19       0.14
Routing Engine status:
  Slot 1:
    Current state                  Present
```

To verify SFM status, use the show chassis sfm command. Sample command output from an M160 router is listed below.

```
user@host> show chassis sfm
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            35      1          0        64        16         46
  1  Empty              0      0          0         0         0          0
  2  Online            47      1          0        64        16         45
  3  Online            50      1          0        64        16         45

Packet scheduling mode : Disabled
```

Table 54 provides an example of jnxFruContent objects in the jnxFruTable for an M40 router.

Table 54: jnxFruContents Objects in the jnxFruTable of an M40 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntryModule	1	online	0	none	0:0:00:00.00	0:0:00:00.00	101974
2	2	0	0	Power supply B	powerEntryModule	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
3	1	0	0	Top impeller	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
3	2	0	0	Bottom impeller	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	0	0	Rear left fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Rear center fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear right fan	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
5	1	0	0	Host controller	routingEngine	1	online	37	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SCB Internet Processor I	controlBoard	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	28	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	29	none	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	online	27	none	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	7	0	0	FPC: 1x OC-48 SONET, SMIR @ 6/*/*	flexiblePicConcentrator	7	online	28	none	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: 1x G/E, 1000 BASE-SX @ 0/0/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	2	0	PIC: 1x Tunnel @ 0/1/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x T1, RJ48 @ 0/2/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x COC12, SMIR @ 0/3/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 2x OC-3 ATM, MM @ 1/0/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x T3 @ 1/2/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x CSTM1, SMIR @ 1/3/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: 4x F/E, 100 BASE-TX @ 3/1/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 1x 800M Crypto @ 3/2/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: 1x CT3-NxDS 0 @ 3/3/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: 1x OC-48 SONET, SMIR @ 6/0/*	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	online	0	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M40 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis
Backplane        REV 03   710-000073   AA2005
Power Supply A   Rev A    740-000235   000119            DC
Maxicab          REV 04   710-000229   AA0691
Minicab          REV 02   710-000482   AA0270
Display          REV 06   710-000150   AA1042
Routing Engine                                          RE-1.0
SCB              REV 07   710-000075   AA1033            Internet Processor I
FPC 0            REV 01   710-001292   AB8159
  PIC 0          REV 08   750-001072   AP5525            1x G/E, 1000 BASE-SX
  PIC 1          REV 01   750-001323   AB1645            1x Tunnel
  PIC 2          REV 01   750-002953   AD9083            4x T1, RJ48
  PIC 3          REV 03   750-001190   AE2907            1x COC12, SMIR
FPC 1            REV 10   710-000175   AA7219
  PIC 0          REV 03   750-002977   HD9331            2x OC-3 ATM, MM
  PIC 1          REV 04   750-002971   HC8020            4x OC-3 SONET, MM
  PIC 2          REV 02.1 710-000608   AA1592            2x T3
  PIC 3          REV 05   750-003248   AD9648            1x CSTM1, SMIR
FPC 3            REV 10   710-000175   AA4782
  PIC 1          REV 04   750-002992   HC3974            4x F/E, 100 BASE-TX
  PIC 2          REV 03   750-003844   AY4806            1x 800M Crypto
  PIC 3          REV 03   750-004743   BD9433            1x CT3-NxDS0
FPC 4            REV 01   710-001292   AC5265
FPC 6            REV 01   710-001292   AB7485
  PIC 0          REV 03   750-000617   AA4566            1x OC-48 SONET, SMIR

user@host> show chassis environment
Class Item                 Status     Measurement
Power Power Supply A       OK
      Power Supply B       Absent
Temp  FPC 0                OK         28 degrees C / 82 degrees F
      FPC 1                OK         29 degrees C / 84 degrees F
      FPC 3                OK         24 degrees C / 75 degrees F
      FPC 4                OK         27 degrees C / 80 degrees F
      FPC 6                OK         28 degrees C / 82 degrees F
      SCB                  OK         27 degrees C / 80 degrees F
      Backplane @ A1       OK         30 degrees C / 86 degrees F
      Backplane @ A2       OK         26 degrees C / 78 degrees F
      Routing Engine       OK         37 degrees C / 98 degrees F
Fans  Top Impeller         OK         Spinning at normal speed
      Bottom impeller      OK         Spinning at normal speed
      Rear Left Fan        OK         Spinning at normal speed
      Rear Center Fan      OK         Spinning at normal speed
      Rear Right Fan       OK         Spinning at normal speed
Misc  Craft Interface      OK
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M40 router is listed below.

```
user@host> show chassis fpc
                    Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            28      2          0         8        11         14
  1  Online            29      7          0         8        21         14
  2  Empty              0      0          0         0         0          0
  3  Online            24     17          0         8        22         15
  4  Online            27      1          0         8         6         13
  5  Empty              0      0          0         0         0          0
  6  Online            28      1          0         8         7         15
  7  Empty              0      0          0         0         0          0
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M40 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                37 degrees C / 98 degrees F
    DRAM                       256 MB
    Memory utilization         19 percent
    CPU utilization:
      User                      1 percent
      Background                0 percent
      Kernel                    3 percent
      Interrupt                 1 percent
      Idle                     96 percent
    Model                      RE-1.0
    Start time                    2002-06-24 17:28:30 UTC
    Uptime                        20 minutes, 30 seconds
    Load averages:                1 minute   5 minute  15 minute
                                      0.00       0.04       0.11
```

To verify SCB status, use the show chassis scb command. Sample command output from an M40 router is listed below.

```
user@host> show chassis scb
SCB status:
  Temperature                  27 degrees C / 80 degrees F
  CPU utilization              3 percent
  Interrupt utilization        0 percent
  Heap utilization             9 percent
  Buffer utilization           44 percent
  Total CPU DRAM               64 MB
  Internet Processor I         Version 1, Foundry IBM, Part number 3
  Start time:                  2002-06-24 17:30:10 UTC
  Uptime:                      19 minutes, 8 seconds
```

Table 55 provides an example of jnxFruContent objects in the jnxFruTable for an M40e router.

Table 55: JnxFruContents Objects in the jnxFruTable of an M40e Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	present	0	none	0:0:00:00.00	0:0:00:25.99	208927
2	2		0	PEM 1	powerEntryModule	2	online	0	none	0:0:00:00.00	0:0:00:25.99	208928
4	1	0	0	Front top blower	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Fan tray front left	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Fan tray front right	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Fan tray rear left	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Fan tray rear right	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear top blower	fan	3	present		none	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Rear bottom blower	fan	4	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	1	0	SFM 0 SPP	switchingAndForwardingModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	2	0	SFM 0 SPR	switchingAndForwardingModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAndForwardingModule	2	online	42	none	0:0:00:21.95	0:0:00:00.00	0
6	2	2	0	SFM 1 SPR Internet Processor II	switchingAndForwardingModule	2	online	42	none	0:0:00:21.95	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	41	none	0:0:00:21.85	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	online	43	none	0:0:00:21.87	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	38	none	0:0:00:21.89	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	online	46	none	0:0:00:21.91	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePicConcentrator	7	empty	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	online	44	none	0:0:00:21.93	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x OC-12 SONET, MM @ 0/1/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x CT3 @ 0/2/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x Multi Link(32) @ 0/3/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: 1x OC-12 SONET, MM @ 2/0/*	portInterfaceCard	3	online	41	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: 1x OC-12 SONET, MM @ 2/1/*	portInterfaceCard	3	online	41	none	0:0:00:21.85	0:0:00:00.00	0
8	3	3	0	PIC: 1x OC-12 SONET, MM @ 2/2/*	portInterfaceCard	3	online	41		0:0:00:21.85	0:0:00:00.00	
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	online	41		0:0:00:21.85	0:0:00:00.00	
8	4	1	0	PIC: 1x OC-48 SONET, SMIR @ 3/0/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	3	0	PIC: 1x OC-12 SONET, SMIR @ 5/2/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	4	0	PIC: 1x OC-12 SONET, MM @ 5/3/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: 8x FE-FX, 100 BASE-FX @ 7/0/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	3	0	PIC: 1x Link Service(4) @ 7/2/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	1	online	46		0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	2	online	46		0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	1	present	34		0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
10	1	1	0	FPM CMB	frontPanelModule	1	online	28		0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanelModule	1	online	28		0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	PCG 0	clockGenerator	1	online	45		0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	PCG 1	clockGenerator	2	online	50		0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	MCS 0	controlBoard	1	online	46		0:0:00:00.00	0:0:00:00.00	0
12	2	0	0	MCS 1	controlBoard	2	online	0		0:0:00:00.00	0:0:00:00.00	0
13	1	0	0	CIP	craftInterfacePanel	1	present	0		0:0:00:00.00	0:0:00:00.00	0

To verify L1, L2, and L3 indexes, use the following commands (M40e example):

```
user@host> show chassis hardware
Item             Version  Part number  Serial number     Description
Chassis                               19084             M40e
Midplane         REV 01   710-005071   AX3654
FPM CMB          REV 03   710-001642   AR9037
FPM Display      REV 03   710-001647   AP1334
CIP              REV 08   710-001593   AE8486
PEM 0            Rev 01   740-003787   ME13120           Power Entry Module
PEM 1            Rev 01   740-003787   MC25354           Power Entry Module
PCG 0            REV 07   710-001568   AG1377
PCG 1            REV 07   710-001568   AR3806
Routing Engine 0 REV 04   740-003239   9001026568        RE-2.0
Routing Engine 1
MCS 0            REV 11   710-001226   AN5810
MCS 1            REV 11   710-001226   AR0109
SFM 1 SPP        REV 07   710-001228   BE0106
SFM 1 SPR        REV 05   710-002189   BE0062            Internet Processor II
FPC 0            REV 01   710-005078   BE0642            M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2496
  PIC 1          REV 04   750-001895   HE0885            1x OC-12 SONET, MM
  PIC 2          REV 06   750-003009   HE1422            4x CT3
  PIC 3          REV 03   750-003837   AP7134            1x Multi Link(32)
FPC 2            REV 01   710-005078   BE0647            M40e-FPC Type 1
  CPU            REV 01   710-004600   AN4299
  PIC 0          REV 04   750-001895   HD2623            1x OC-12 SONET, MM
  PIC 1          REV 04   750-001895   HE0609            1x OC-12 SONET, MM
  PIC 2          REV 04   750-001895   HE0871            1x OC-12 SONET, MM
FPC 3            REV 01   710-005197   BD9846            M40e-FPC Type 2
  CPU            REV 01   710-004600   BD2364
  PIC 0          REV 01   750-001900   AA9649            1x OC-48 SONET, SMIR
FPC 5            REV 01   710-005078   BE0639            M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2587
  PIC 2          REV 04   750-001896   AV4480            1x OC-12 SONET, SMIR
  PIC 3          REV 04   750-001895   HE1000            1x OC-12 SONET, MM
FPC 7            REV 01   710-005196   BD9456            M40e-FPC
  CPU            REV 01   710-004600   AN4323
  PIC 0          REV 01   750-004944   AY4645            8x FE-FX, 100 BASE-FX
  PIC 2          REV 01   750-007927   AP1919            1x Link Service(4)
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M40e router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                Master
    Election priority            Master (default)
    Temperature                   34 degrees C / 93 degrees F
    DRAM                         768 MB
    Memory utilization           9 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     2 percent
      Interrupt                  0 percent
      Idle                       97 percent
    Model                          RE-2.0
    Serial ID                      9c000007c8644701
    Start time                     2002-06-24 10:33:41 PDT
    Uptime                         31 minutes, 7 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.01       0.02       0.00
Routing Engine status:
  Slot 1:
    Current state                  Present
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M40e router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            41      4          0        32         3         40
  1  Empty              0      0          0         0         0          0
  2  Online            43      4          0        32         1         40
  3  Online            38      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Online            46      4          0        32         1         40
  6  Empty              0      0          0         0         0          0
  7  Online            44      4          0        32         2         39
```

Table 56 provides an example of jnxFruContent objects in the jnxFruTable for a T640 routing node.

Table 56: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntry-Module	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	powerEntry-Module	2	online	27	none	0:0:00:00.00	0:0:00:00.00	217044
4	1	1	0	Top left front fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Top left middle fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top left rear fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Top right front fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	5	0	Top right middle fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	6	0	Top right rear fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	6	0	Bottom right rear fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	1	0	Fourth blower from top	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	2	0	Bottom blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	3	0	Middle blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	3	4	0	Top blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	5	0	Second blower from top	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePic-Concentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	1	0	FPC @ 0/0/* top temp. sensor	flexiblePic-Concentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	2	0	FPC @ 0/1/* bottom temp. sensor	flexiblePic-Concentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePic-Concentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0
7	2	1	0	FPC @ 1/0/* top temp. sensor	flexiblePic-Concentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0
7	2	2	0	FPC @ 1/1/* bottom temp. sensor	flexiblePic-Concentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePic-Concentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	3	1	0	FPC @ 2/0/* top temp. sensor	flexiblePic-Concentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	3	2	0	FPC @ 2/1/* bottom temp. sensor	flexiblePic-Concentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePic-Concentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	1	0	FPC @ 3/0/* top temp. sensor	flexiblePic-Concentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	2	0	FPC @ 3/1/* bottom temp. sensor	flexiblePic-Concentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	5	9	0	FPC @ 4/*/*	flexiblePic-Concentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	5	1	0	FPC @ 4/0/* top temp. sensor	flexiblePic-Concentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0
7	5	2	0	FPC @ 4/1/* bottom temp. sensor	flexiblePic-Concentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePic-Concentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	6	1	0	FPC @ 5/0/* top temp. sensor	flexiblePic-Concentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	6	2	0	FPC @ 5/1/* bottom temp. sensor	flexiblePic-Concentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePic-Concentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	7	1	0	FPC @ 6/0/* top temp. sensor	flexiblePic-Concentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	7	2	0	FPC @ 6/1/* bottom temp. sensor	flexiblePic-Concentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePic-Concentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	8	1	0	FPC @ 7/0/* top temp. sensor	flexiblePic-Concentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	8	2	0	FPC @ 7/1/* bottom temp. sensor	flexiblePic-Concentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterface-Card	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterface-Card	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterface-Card	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	4	0	PIC: @ 0/3/*	portInterface-Card	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x OC-48 SONET, SMIR @ 1/0/*	portInterface-Card	2	online		none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 1x OC-48 SONET, SMSR @ 1/1/*	portInterface-Card	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 1x OC-48 SONET, SMIR @ 1/2/*	portInterface-Card	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x OC-48 SONET, SMIR @ 1/3/*	portInterface-Card	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterface-Card	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterface-Card	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterface-Card	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterface-Card	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterface-Card	4	online		none	0:0:00:01.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterface-Card	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterface-Card	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterface-Card	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	5	1	0	PIC: 1x Tunnel @ 4/0/*	portInterface-Card	5	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	5	2	0	PIC: 1x OC-192 SM SR2 @ 4/1/*	portInterface-Card	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0
8	5	3	0	PIC: 4x OC-48 SONET, SMSR @ 4/2/*	portInterface-Card	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	4	0	PIC: 1x OC-192 SM SR1 @ 4/3/*	portInterface-Card	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterface-Card	6	empty	0	none	0:0:00:01.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterface-Card	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterface-Card	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterface-Card	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterface-Card	7	online	30	none	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterface-Card	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterface-Card	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterface-Card	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterface-Card	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterface-Card	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterface-Card	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterface-Card	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routing-Engine	1	online	34	none	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routing-Engine	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
10	1	1	0	FPM GBUS	frontPanel-Module	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanel-Module	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	clockGener-ator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	clockGener-ator	2	online	27	none	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	control-Board	1	online	27	none	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	control-Board	2	unkno wn	0	none	0:0:00:01.96	0:0:00:00.00	0
13	1	0	0	CIP	craftInter-facePanel	1	present	36	none	0:0:00:00.00	0:0:00:00.00	0
14	1	0	0	SPMB 0	processor-Mezzanine-BoardForSIB	1	online	34	none	0:0:00:00.00	0:0:00:00.00	0
14	2	0	0	SPMB 1	processor-Mezzanine-BoardForSIB	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	1	0	0	SIB 0	switchInter-faceBoard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	switchInter-faceBoard	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	switchInter-faceBoard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	switchInter-faceBoard	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	switchInter-faceBoard	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                1234              T640
Midplane         REV 04   710-002726   AX5603
FPM GBUS         REV 02   710-002901   HE3062
FPM Display      REV 01   710-002897   HD3033
CIP              REV 05   710-002895   HA5022
PEM 1            RevX02   740-002595   MD21812           Power Entry Module
SCG 1            REV 01   710-003423   HD3025
Routing Engine 0 REV 01   740-005022   210865700336      RE-3.0
CB 0             REV 02   710-002728   HE3025
CB 1
FPC 1            REV 01   710-002385   HE3173            FPC Type 2
  CPU            REV 06   710-001726   HC0042
  PIC 0          REV 03   750-001900   AD5737            1x OC-48 SONET, SMIR
  PIC 1          REV 07   750-001900   AR3613            1x OC-48 SONET, SMSR
  PIC 2          REV 01   750-001900   AA9604            1x OC-48 SONET, SMIR
  PIC 3          REV 01   750-001900   AA9602            1x OC-48 SONET, SMIR
  MMB 1          REV 03   710-001723   HC0111            MMB-144mbit
  ICBM           REV 04   710-003384   HA4497
  PPB 0          REV 02   710-003758   HA4543            PPB Type 2
  PPB 1          REV 02   710-003758   HA4540            PPB Type 2
FPC 2            REV 01   710-002385   HE3180            FPC Type 2
  CPU            REV 06   710-001726   HE7904
  MMB 1          REV 03   710-001723   HC0120            MMB-144mbit
  ICBM           REV 01   710-003384   HE3046
  PPB 0          REV 02   710-003758   HA4564            PPB Type 2
  PPB 1          REV 02   710-003758   HA4554            PPB Type 2
FPC 4            REV 04   710-001721   HE3145            FPC Type 3
  CPU            REV 06   710-001726   HC0034
  PIC 0                                                  1x Tunnel
  PIC 1          REV 01   750-003824   HE7803            1x OC-192 SM SR2
  PIC 2          REV 01   750-003336   HE3420            4x OC-48 SONET, SMSR
  PIC 3          REV 01   750-003824   HE7802            1x OC-192 SM SR1
  MMB 0          REV 03   710-001723   HE7230            MMB-144mbit
  MMB 1          REV 03   710-001723   HE7267            MMB-144mbit
  ICBM           REV 04   710-003384   HA4485
  PPB 0          REV 02   710-002845   HA4550            PPB Type 3
  PPB 1          REV 02   710-002845   HA4525            PPB Type 3
FPC 5            REV 04   710-001721   HE3175            FPC Type 3
  CPU
FPC 6            REV 01   710-002385   HD5027            FPC Type 2
  CPU            REV 06   710-001726   HC0033
  MMB 1          REV 03   710-001723   HC0080            MMB-144mbit
  ICBM           REV 04   710-003384   HA4486
  PPB 0          REV 02   710-003758   HA4541            PPB Type 2
  PPB 1          REV 02   710-003758   HA4539            PPB Type 2
SPMB 0           REV 01   710-003229   HA5999
SIB 0            REV 01   710-003980   HD5054            SIB-I8
SIB 2            REV 01   710-003980   HC0035            SIB-I8
SIB 3            REV 01   710-003980   HA5065            SIB-I8
SIB 4            REV 01   710-003980   HE3016            SIB-I8
```

To verify FPC status, use the show chassis fpc command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty              0      0          0         0         0          0
  1  Online            30      2          0       512         3         41
  2  Online            30      2          0       256         7         41
  3  Empty              0      0          0         0         0          0
  4  Online            30      4          0       512         6         41
  5  Offline         --- Unresponsive ---
  6  Online            30      2          0       256         7         41
  7  Empty              0      0          0         0         0          0
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                      2048 MB
    Memory utilization           4 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     2 percent
      Interrupt                  0 percent
      Idle                      97 percent
    Model                        RE-3.0
    Start time                   2002-06-24 10:33:34 PDT
    Uptime                       33 minutes, 38 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.08       0.03       0.01
```

To verify SPMB status, use the show chassis spmb command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis spmb
Slot 0 information:
  State                          Online
  Total CPU Utilization             2%
  Interrupt CPU Utilization         0%
  Memory Heap Utilization           0%
  Buffer Utilization               40%
  Start time:                    2002-06-24 10:34:22 PDT
  Uptime:                         33 minutes, 3 seconds
```

Chassis Traps

The chassis-related traps are defined under the jnxTraps and jnxChassisOKtraps branches. For the system logging severity levels for these traps, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 111.

These traps are defined as follows:

- Power failure (jnxPowerSupplyFailure)—When the power supply, router circuit breaker, or power circuit fails, or when there is a power outage. If only one of the power supplies in the router fails, the service impact is minimal. One power supply can provide the necessary power for a fully loaded router. To determine the source of the failure, you must physically inspect the router.

- Fan failure (jnxFanFailure)—When the fan fuse blows or when the fan wiring shorts out. If only one fan has failed, there is no service impact. The remaining fans increase speed to compensate. However, you must resolve the problem before another fan fails. To determine the source of the failure, you must physically inspect the router, taking care to check the fuses. See the hardware installation guide for your router model for more information.

- Overtemperature (jnxOverTemperature)—When several fans fail or the room temperature increases significantly. The service impact of this trap depends on the temperature of the router. In general, the router increases the speed of the fans when any component exceeds a temperature of 55 °C. The fans remain at the higher speed until the temperature decreases below the threshold. In this case, there is no service impact. However, if the temperature exceeds 75 °C, the router transmits a warning and automatically shuts down. This scenario creates a significant service impact because the shutdown affects additional routers and equipment. To determine the source of the overtemperature problem, you must physically inspect the router to determine whether any fans have failed in the router.

- Power Supply OK (jnxPowerSupplyOK)—Sent when a power supply recovers from failure.

- Fan OK (jnxFanOK)—Sent when a fan recovers from failure.

- Temperature OK (jnxTemperatureOK)—Sent when a chassis component recovers from an overtemperature condition.

- Redundancy Switchover (jnxRedundancySwitchover)—For certain platforms, such as the M20 or M160, some subsystems, such as the Routing Engine, have a redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions. The redundancy switchover trap indicates such a change.

- Field Replaceable Unit Removal (jnxFruRemoval)—Sent when the specified FRU has been removed from the chassis.

- Field Replaceable Unit Insertion (jnxFruInsertion)—Sent when the specified FRU has been inserted into the chassis.

- Field Replaceable Unit Power Off (jnxFrulPoweroff)—Sent when the specified FRU has been powered off in the chassis.
- Field Replaceable Unit Power On (jnxFruPowerOn)—Sent when the specified FRU has been powered on in the chassis.
- Field Replaceable Unit Failed (jnxFruFailed)—Sent when the specified FRU has failed in the chassis. Typically, this is due to the FRU not powering up or being unable to load software. FRU replacement may be required.
- Field Replaceable Unit Offline (jnxFruOffline)—Sent when the specified FRU goes offline.
- Field Replaceable Unit Online (jnxFruOnline)—Sent when the specified FRU goes online.
- Field Replaceable Unit Check (jnxFruCheck)—Sent when the specified FRU has encountered operational errors.

For more information on chassis MIB traps, see “Standard SNMP Traps” on page 119 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 111.

This section contains the following topics:

SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in Table 57. To view the SNMPv1 chassis-related traps, see "Standard SNMP Traps" on page 119 and "Juniper Networks Enterprise-Specific SNMP Traps" on page 111.

The column headings describe the SNMPv1 traps format:

- Trap Name—The name of the trap.
- Enterprise ID—The identification number of the enterprise-specific trap.
- Generic Trap Number—The generic trap number field of the SNMP trap PDU. This field is enterpriseSpecific(6) for enterprise-specific traps, other predefined values for standard traps.
- Specific Trap Number— The specific trap number field of the SNMP trap PDU. For standard traps, this field is zero; for enterprise-specific traps, this field is nonzero as defined in the enterprise-specific MIBs.

Table 57: SNMP Version 1 Trap Format

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2
jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2
jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12
jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9
jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6
jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10
jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8
jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5
jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2	6	1
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1	6	4
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3

SNMPv2 Trap Format

The SNMPv2 trap format for the chassis MIB traps is described in Table 58.

The column headings describe the SNMPv2 traps format:

- Trap Name—The name of the trap.
- snmpTrapOID—The authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2 trap PDU and InformRequest PDU.
- Description—The JUNOS enterprise-specific name of the trap.

Table 58: SNMP Version 2 Trap Format

Trap Name	snmpTrapOID	Description
jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	The fan fuse blows or the fan wiring shorts out.
jnxFanOK	1.3.6.1.4.1.2636.4.2.2	The fan has recovered from failure state.
jnxFruCheck	1.3.6.1.4.1.2636.4.1.12	The FRU has operational errors and has gone into a self-check diagnosis state.
jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	The FRU has been inserted into the chassis.
jnxFruFailed	1.3.6.1.4.1.2636.4.1.9	The FRU has failed in the chassis.
jnxFruOffline	1.3.6.1.4.1.2636.4.1.10	The FRU has gone offline.
jnxFruOnline	1.3.6.1.4.1.2636.4.1.11	The FRU has gone back online.
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	The FRU has been powered off in the chassis.
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	The FRU has been powered on in the chassis.
jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	The FRU has been removed from the chassis.
jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	Several fans fail or the room temperature increases significantly.
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	The power supply, router circuit breaker, or power circuit fails, or there is a power outage.
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	The power supply has recovered from failure state.
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1.4	A redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions.
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	The component sensor has detected the overtemperature condition.

Chassis Definitions for the Router Model MIB

The enterprise-specific chassis definitions for the router model MIB contains the OIDs that are used by the chassis MIB to identify platform and chassis components. The chassis MIB provides information that changes often. The chassis definitions for the router model MIB provides information that changes less often.

The last number in each sysObjectId, shown in Table 59, corresponds to the router model and therefore does not change.

Table 59: Router Models and Their sysObjectIds

Model	SysObjectID	jnxProductName
J2300	1.3.6.1.4.1.2636.1.1.1.2.13	jnxProductNameJ2300
J4300	1.3.6.1.4.1.2636.1.1.1.2.14	jnxProductNameJ4300
J6300	1.3.6.1.4.1.2636.1.1.1.2.15	jnxProductNameJ6300
M5	1.3.6.1.4.1.2636.1.1.1.2.5	jnxProductNameM5
M7i	1.3.6.1.4.1.2636.1.1.1.2.10	jnxProductNameM7i
M10	1.3.6.1.4.1.2636.1.1.1.2.4	jnxProductNameM10
M10i	1.3.6.1.4.1.2636.1.1.1.2.11	jnxProductNameM10i
M20	1.3.6.1.4.1.2636.1.1.1.2.2	jnxProductNameM20
M40	1.3.6.1.4.1.2636.1.1.1.2.1	jnxProductNameM40
M40e	1.3.6.1.4.1.2636.1.1.1.2.8	jnxProductNameM40e
M160	1.3.6.1.4.1.2636.1.1.1.2.3	jnxProductNameM160
M320	1.3.6.1.4.1.2636.1.1.1.2.9	jnxProductNameM320
TX	1.3.6.1.4.1.2636.1.1.1.2.17	jnxProductNameTX
T320	1.3.6.1.4.1.2636.1.1.1.2.7	jnxProductNameT320
T640	1.3.6.1.4.1.2636.1.1.1.2.6	jnxProductNameT640

For a downloadable version of the chassis definitions for the router model MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-chas-defines.txt.

Chapter 20

Interpreting the Enterprise-Specific Destination Class Usage MIB

The enterprise-specific destination class usage (DCU) Management Information Base (MIB) counts packets from customers by performing a lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

The DCU MIB is a subbranch of the jnxMibs branch of the enterprise-specific MIB {enterprise 2636} and has an object identifier of {jnxMIB 6}. The DCU MIB has one branch, jnxDCUs, which contains two tables: jnxDCUsTable and jnxDcuStatsTable. For information about configuring source and destination class usage, see the *JUNOS Policy Framework Configuration Guide* and *JUNOS Network Interfaces Configuration Guide*. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-dcu.txt.

This chapter contains the following topics:

jnxDCUsTable

The entries in the jnxDCUsTable, whose object identifier is {jnxDCUTable 1}, are represented by jnxDCUsEntry and are listed in Table 60.

Table 60: jnxDCUsEntry

Object	Object Identifier	Description
jnxDCUSrcIfIndex	jnxDUCsEntry 1	The interface index of the ingress interface
jnxDCUDstClassName	jnxDCUsEntry 2	The destination class name specified in a routing policy and applied to the forwarding table.
jnxDCUPackets	jnxDCUsEntry 3	The number of packets passing through the network.
jnxDCUBytes	jnxDCUsEntry 4	The number of bytes passing through the network.

jnxDcuStatsTable

jnxDcuStatsTable contains statistics for traffic that satisfies the rules in each configured destination class. A separate set of statistics is kept for each destination class on each interface and address family on which this feature is enabled. This is essentially a replacement for jnxDCUsTable.

The entries in the jnxDcuStatsTable, whose object identifier is {jnxDCUs 2}, are represented by jnxDCUsStatusEntry and are listed in Table 61.

Table 61: jnxDCUsStatusEntry

Ojbect	Object Identifier	Description
jnxDcuStatsSrcIfIndex	jnxDcuStatsEntry 1	The interface index of the ingress interface for traffic counted in each entry.
jnxDcuStatsAddrFamily	jnxDcuStatsEntry 2	The address family of the entry's traffic.
nxDcuStatsClassName	jnxDcuStatsEntry 3	The name of the destination class that applies to the entry's traffic.
jnxDcuStatsPackets	jnxDcuStatsEntry 4	The number of packets received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsBytes	jnxDcuStatsEntry 5	The number of bytes received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsClName	jnxDcuStatsEntry 6	The name of the destination class. This object is a duplicate of jnxDcuStatsClassName and is included to satisfy those network management applications that cannot extract the destination class name from the instance portion of the OID.

Chapter 21

Interpreting the Enterprise-Specific BGP4 V2 MIB

The enterprise-specific Border Gateway Protocol version 4 (BGP4) V2 MIB, whose object identifier is {jnxBgpM2Experiment 1}, contains objects used to monitor BGP peer-received prefix counters. It is based upon similar objects in the MIB documented in *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version*, Internet draft draft-ietf-idr-bgp4-mibv2-03.txt. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-bgpmib2.txt.

NOTE: For the BGP4 V2 MIB, the JUNOS software supports only the following objects: jnxBgpM2PrefixInPrefixes, jnxBgpM2PrefixInPrefixesAccepted, and jnxBgpM2PrefixInPrefixesRejected.

This chapter discusses the following topic:

jnxBgpM2PrefixCountersTable

jnxBgpM2PrefixCountersTable contains counters associated with a BGP peer.

JnxBgpM2PrefixCountersEntry

jnxBgpM2PrefixCountersEntry contains information about the prefix counters of a BGP peer, and the objects listed in Table 62.

Table 62: jnxVpnInfo

Object	Object Identifier	Description
jnxBgpM2PrefixInPrefixes	jnxBgpM2PrefixCountersEntry 7	The total number of prefixes received from a peer.
jnxBgpM2PrefixInPrefixesAccepted	jnxBgpM2PrefixCountersEntry 8	The total number of prefixes received from a peer that are eligible to be active in the routing table.
jnxBgpM2PrefixInPrefixesRejected	jnxBgpM2PrefixCountersEntry 9	The total number of prefixes received from a peer that are not eligible to be active in the routing table.

Chapter 22

Interpreting the Enterprise-Specific Ping MIB

The ping MIB extends the standard ping MIB control table (RFC 2925). The ping MIB, whose object identifier is jnxMIbs 7, allows you to monitor network delay (latency), packet loss, network delay variation (jitter), one-way latency, and other network statistics.

Items in this MIB are created when entries are created in the pingCtlTable of the ping MIB. Each item is indexed exactly as in the ping MIB.

To view a complete copy of the enterprise-specific extensions to the ping MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-ping.txt. For more information on using the ping MIB and enterprise-specific ping MIB, see "SNMP Remote Operations" on page 83. For information about how to configure thresholds at the [edit services rpm] hierarchy level, see the *JUNOS Services Interfaces Configuration Guide*.

This section includes the following topics:

jnxPingCtlTable

The enterprise-specific ping MIB structure includes one main object, jnxPingCtlTable, whose object identifier is {jnxPingObjects 2}, and defines the jnxPing control table for providing enterprise-specific options to the corresponding pingCtlEntry. jnxpingCtTable monitors thresholds; for example, the maximum allowed jitter in the trip time during a text.

jnxPingCtlEntry

Each jnxPingCtlEntry has two indexes identical to those of the corresponding pingCtlEntry. Entries created in pingCtlTable are mirrored here. jnxPingCtlEntry objects are listed in the Table 63.

Table 63: jnxPingCtlEntry

Object	Object Identifier	Description
jnxCtlOwnerIndex	jnxPingCtlEntry 1	The first index. It is identical to the pingCtlOwnerIndex of the corresponding pingCtlEntry in the pingCtlTable.
jnxPingCtlTestName	jnxPingCtlEntry 2	The other index and is identical to the pingCtlTestName of the corresponding pingCtlEntry in the pingCtlTable.
jnxPingCtlIfName	jnxPingCtlEntry 3	Specifies the name of the outgoing interface for ping probes. This is the name-based complement to pingCtlIfIndex. A zero-length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: ■ pingCtlIfIndex (see pingCtlTable in the ping MIB) ■ jnxPingCtlIfName ■ jnxPingCtlRoutingInstanceName
jnxPingCtlRoutingInstanceName	jnxPingCtlEntry 6	Specifies the name of the routing instance used when directing outgoing ping packets. The instance name specified must be configured at the [edit routing-instances] hierarchy level of the JUNOS configuration. The instance-type must be vrf.
jnxPingCtlRttThreshold	jnxPingCtlEntry 7	The maximum round-trip time allowed. If this threshold is crossed by any probe, a jnxPingRttThresholdExceeded trap will be sent.
jnxPingCtlRttStdDevThreshold	jnxPingCtlEntry 8	The maximum round-trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the round-trip time at the end of any test exceeds this threshold, a jnxPingRttStdDevThresholdExceeded trap will be sent.
jnxPingCtlRttJitterThreshold	jnxPingCtlEntry 9	The maximum allowed jitter in the round-trip time over the course of any test. Jitter is the difference between the maximum and minimum round-trip times measured over the course of a single test (jnxPingResultsMaxRttUs minus jnxPingResultsMinRttUs). If the measured jitter exceeds this threshold, a jnxPingRttJitterThresholdExceeded trap is sent.
jnxPingCtlEgressTimeThreshold	jnxPingCtlEntry 10	Maximum egress trip time allowed. If this threshold is crossed by any probe, a jnxPingEgressThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. Currently jnxPingIcmpTimeStamp is the only supported probe type with this property.

Object	Object Identifier	Description
jnxPingCtlEgressStdDevThreshold	jnxPingCtlEntry 11	The maximum egress trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the egress trip time at the end of any test exceeds this threshold, a jnxPingEgressStdDevThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. The jnxPingIcmpTimeStamp is the only supported probe type with this property.
jnxPingCtlEgressJitterThreshold	jnxPingCtlEntry 12	The maximum allowed jitter in the egress trip time over the course of any test. Jitter is defined as the difference between the maximum and minimum egress trip times measured over the course of a single test (jnxPingResultsMaxSrcDstt minus jnxPingResultsMinSrcDstt). If the measured jitter exceeds this threshold, a jnxPingEgressJitterThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. The jnxPingIcmpTimeStamp is the only supported probe type with this property.
jnxPingCtlIngressTimeThreshold	jnxPingCtlEntry 13	The maximum ingress trip time allowed. If this threshold is crossed by any probe, a jnxPingIngressThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. The jnxPingIcmpTimeStamp is the only supported probe type with this property.
jnxPingCtlIngressStddevThreshold	jnxPingCtlEntry 14	The maximum ingress trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the ingress trip time at the end of any test exceeds this threshold, a jnxPingIngressStddevThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. Currently jnxPingIcmpTimeStamp is the only supported probe type with this property.
jnxPingCtlIngressJitterThreshold	jnxPingCtlEntry 15	The maximum allowed jitter in the ingress trip time over the course of any test. Jitter is defined as the difference between the maximum and minimum ingress trip times measured over the course of a single test (jnxPingResultsMaxDstSrct minus jnxPingResultsMinDstSrct). If the measured jitter exceeds this threshold, a jnxPingIngressJitterThresholdExceeded trap will be sent. This applies only if the probe type (pingCtlType) provides one-way delay measurements. The jnxPingIcmpTimeStamp is the only supported probe type with this property.

Object	Object Identifier	Description
jnxPingCtlTrapGeneration	jnxPingCtlEntry 16	The value of this object determines when and if to generate a notification for this entry. rttThreshold(0)—Generate a jnxPingRttThresholdExceeded notification when the configured rtt threshold is exceeded. rttStdDevThreshold(1)—Generate a jnxPingRttStdDevThresholdExceeded notification when the configured rtt standard deviation threshold is exceeded. rttJitterThreshold(2)—Generate a jnxPingRttJitterThresholdExceeded notification when the configured rtt jitter threshold is exceeded. egressThreshold(3)—Generate a jnxPingEgressThresholdExceeded notification when the configured egress threshold is exceeded. This applies only if the probe type supports one-way measurements.
		egressStdDevThreshold(4)—Generate a jnxPingEgressStdDevThresholdExceeded notification when the configured egress standard deviation threshold is exceeded. This applies only if the probe type supports one-way measurements. egressJitterThreshold(5)—Generate a jnxPingEgressJitterThresholdExceeded notification when the configured egress jitter threshold is exceeded. This applies only if the probe type supports one-way measurements. ingressThreshold(6)—Generate a jnxPingIngressThresholdExceeded notification when the configured ingress threshold is exceeded. This applies only if the probe type supports one-way measurements. ingressStdDevThreshold(7)—Generate a jnxPingIngressStdDevThresholdExceeded notification when the configured ingress standard deviation threshold is exceeded. This applies only if the probe type supports one way measurements. ingressJitterThreshold(8)—Generate a jnxPingIngressJitterThresholdExceeded notification when the configured ingress jitter threshold is exceeded. This applies only if the probe type supports one-way measurements. The value of this object defaults to zero, indicating that none of the above options have been selected.

jnxPingResultsTable

jnxPingResultsTable, whose object identifier is jnxPingObjects 3, gathers ping test results on traffic on round-trip, ingress, and egress trip delays. This useful when you want to measure the performance of your network and verify service-level agreements with your vendors.

jnxpingResultsEntry

The jnxPingResultsEntry objects are listed in Table 64.

Table 64: jnxPingsResultsEntry

Object	Object Identifier	Description
jnxPingResultsRttUs	jnxPingResultsEntry 1	The round-trip delays measured for the most recent successful probe during this test, in microseconds.
jnxPingResultsSumRttUs	jnxPingResultsEntry 2	The sum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsMinRttUs	jnxPingResultsEntry 3	The minimum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsMaxRttUs	jnxPingResultsEntry 4	The maximum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsAvgRttUs	jnxPingResultsEntry 5	The average of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsStdDevRttUs	jnxPingResultsEntry 6	The standard deviation of the round-trip delays measured during this test, in microseconds.
jnxPingResultsEgressUs	jnxPingResultsEntry 7	The egress trip delays measured for the most recent successful probe during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsMinEgressUs	jnxPingResultsEntry 8	The minimum of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.

Object	Object Identifier	Description
jnxPingResultsMaxEgressUs	jnxPingResultsEntry 9	The maximum of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsAvgEgressUs	jnxPingResultsEntry 10	The average of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsStddevEgressUs	jnxPingResultsEntry 11	The standard deviation of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsIngressUs	jnxPingResultsEntry 12	The ingress trip delays measured for the most recent successful probe during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0."
jnxPingResultsMinIngressUs	jnxPingResultsEntry 13	The minimum of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsMaxIngressUs	jnxPingResultsEntry 14	The maximum of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.

Object	Object Identifier	Description
jnxPingResultsAvgIngressUs	jnxPingResultsEntry 15	The average of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsStddevIngressUs	jnxPingResultsEntry 16	The standard deviation of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsJitterRttUs	jnxPingResultsEntry 17	The jitter of the round-trip delays measured for all probes during this test, in microseconds.
jnxPingResultsJitterEngressUs	jnxPingResultsEntry 18	The jitter of the engress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsJitterIngressUs	jnxPingResultsEntry 19	The jitter of the ingress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingResultsStatus	jnxPingResultsEntry 20	The result of the most recent probe.
jnxPingResultsTime	jnxPingResultsEntry 21	The date and time of the most recent probe result.

Object	Object Identifier	Description
jnxPingResultsOwnerIndex	jnxPingResultsEntry 22	The first index. It has the same value as pingCtlOwnerIndex and is provided for applications that are unable to obtain the value of pingCtlOwnerIndex from the instance portion of the object identifiers belonging to this table.
jnxPingResultsTestName	jnxPingResultsEntry 23	The other index. It has the same value as pingCtlTestName and is provided for applications that are unable to obtain the value of pingCtlTestName from the instance portion of the object identifiers belonging to this table.

jnxPingProbeHistoryTable

jnxpingProbeHistoryTable contains the history of all ping tests.

jnxPingProbeHistoryEntry

The jnxPingProbeHistoryEntry objects are listed in Table 65.

Table 65: jnxPingProbeHistoryEntry

Object	Object Identifier	Description
jnxPingProbeHistoryResponseUs	jnxPingProbeHistoryEntry 1	The amount of time, in microseconds, from when a probe was sent to when its response was received or when it timed out. The value of this object is reported as 0 when it is not possible to transmit a probe.
jnxPingProbeHistoryJitterUs	jnxPingProbeHistoryEntry 2	The time difference, in microseconds, between the maximum and minimum round-trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed.

Object	Object Identifier	Description
jnxPingProbeHistoryResponseEgressUs	jnxPingProbeHistoryEntry 3	The amount of time, in microseconds, from when a probe was sent to when it was received by destination. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.
jnxPingProbeHistoryResponseIngressUs	jnxPingProbeHistoryEntry 4	The amount of time, in microseconds, from when a probe was sent from the destination to when it was received. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.
jnxPingProbeHistoryEgressJitterUs	jnxPingProbeHistoryEntry 5	The time difference, in microseconds, between the maximum and minimum egress trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.
jnxPingProbeHistoryIngressJitterUs	jnxPingProbeHistoryEntry 6	The time difference, in microseconds, between the maximum and minimum ingress trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.

Chapter 23

Interpreting the Enterprise-Specific Traceroute MIB

The traceroute MIB supports the JUNOS software extensions of traceroutes and remote operations. Items in this MIB are created when entries are created in the traceRouteCtlTable of the traceroute MIB. Each item is indexed exactly the same way as it is in the traceroute MIB. For a downloadable version of the traceroute MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-traceroute.txt. For more information on using the traceroute MIB and enterprise-specific traceroute MIB, see "SNMP Remote Operations" on page 83.

The enterprise-specific tracerouteMIB structure includes one main object, jnxTraceRouteCtlTable.

jnxTraceRouteCtlTable

The jnxTraceRouteCtlTable, whose object identifier is {jnxTraceRouteObjects 2}, defines the jnxTraceRoute control table for providing enterprise-specific options to the corresponding traceRouteCtlEntry.

jnxTraceRouteCtlEntry

Each jnxTraceRouteCtlEntry has two indexes that are identical to those of the corresponding TraceRouteCtlEntry. Entries created in TraceRouteCtlTable are mirrored here and are listed in Table 66.

Table 66: jnxTraceRouteCtlTable

Object	Object Identifier	Description
jnxTRCtlOwnerIndex	jnxTraceRouteCtlEntry 1	The first index. It is identical to the jnxTraceRouteCtlOwnerIndex of the corresponding jnxTraceRouteCtlEntry in the jnxTraceRouteCtlTable.
jnxTRCtlTestName	jnxTraceRouteCtlEntry 2	The other index. It is identical to the jnxTraceRouteCtlTestName of the corresponding jnxTraceRouteCtlEntry in the jnxTraceRouteCtlTable.

Object	Object Identifier	Description
jnxTRCtlIfName	jnxTraceRouteCtlEntry 3	Specifies the name of the outgoing interface for traceroute probes. This is the name-based complement to traceRouteCtlIfIndex. A zero-length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: ■ traceRouteCtlIfIndex (see traceRouteCtlTable in the traceroute MIB) ■ jnxTRCtlIfName ■ jnxTRCRoutingInstanceName
jnxTRCtlRoutingInstanceName	jnxTraceRouteCtlEntry 4	Specifies the name of the routing instance used when directing outgoing traceroute packets. The instance name specified must be configured at the [edit routing-instances] hierarchy level of the JUNOS configuration.

Chapter 24

Interpreting the Enterprise-Specific RMON Events and Alarms MIB

The remote monitoring (RMON) events and alarms MIB monitors objects on a device and warns the network system administrator if one of those values exceeds the defined range. The alarm monitors objects in this MIB and triggers an event when the condition (falling or rising threshold) is reached.

The Juniper Networks enterprise-specific extension to the standard RMON MIB augments the alarmTable with additional information about each alarm. Two new traps, jnxRmonAlarmGetFailure and jnxRmonGetOk, are also defined to indicate when problems are encountered with an alarm.

To view a complete copy of the enterprise-specific extensions to the RMON MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-rmon.txt. For more information on RMON alarms and events, see "RMON Alarms and Events" on page 183.

This chapter contains the following topics:

jnxRmonAlarmTable

The entries in the jnxRmonAlarmTable, whose object identifier is {jnxMibs 13}, are represented by jnxRmonAlarmEntry, whose object identifier is {jnxRmonAlarmTable1} and are listed in Table 67.

Table 67: jnxRmonAlarmEntry

Object	Object Identifier	Description
jnxRmonAlarmGetFailCnt	jnxRmonAlarmEntry 1	Represents the number of times the internal Get request for the variable monitored by this entry has failed.
jnxRmonAlarmGetFailTime	jnxRmonAlarmEntry 2	Represents the value of sysUpTime when an internal Get request for the variable monitored by this entry last failed.
jnxRmonAlarmGetFailReason	jnxRmonAlarmEntry 3	■ Represents the reason an internal Get request for the variable monitored by this entry last failed. This object contains the following values: ■ other (1)—An error was encountered that does not fit into one of the currently defined categories. ■ noError (2)—Get request processed successfully. ■ noSuchObject (3)—Requested object not available. ■ outOfView (4)—Requested object instance out of MIB view. ■ noSuchInstance (5)—Requested object instance not available. ■ badReqId (6)—Unexpected request ID encountered while processing Get request. ■ oidMatchErr (7)—Unexpected object ID encountered while processing Get request. ■ oidBindErr (8)—Unable to bind object ID to Get request PDU. ■ createPktErr (9)—Unable to create Get request PDU. ■ badObjType (10)—Unexpected object type encountered while processing Get request.

Object	Object Identifier	Description
jnxRmonAlarmGetOkTime	jnxRmonAlarmEntry 4	Represents the value of sysUpTime when an internal Get request for the variable monitored by this entry succeeded and the entry left the getFailure state.
jnxRmonAlarmState	jnxRmonAlarmEntry 5	Represents the current state of this RMON alarm entry. This object contains the following values: ■ unknown (1)—Alarm entry unknown ■ underCreation (2)—Alarm entry not activated ■ active (3)—Alarm entry active and within thresholds ■ startup (4)—Alarm entry still waiting for first value ■ risingThreshold (5)—Alarm entry has crossed the rising threshold. ■ fallingThreshold (6)—Alarm entry has crossed the falling threshold ■ getFailure (7)—Alarm entry internal Get request failed.

RMON Event and Alarm Traps

The following traps send notifications when there is a problem with RMON alarm processing and are listed in Table 68.

Table 68: RMON Event and Alarm Traps

Trap	Object Identifier	Description
jnxRmonAlarmGetFailure	jnxRmonTrapPrefix 1	Generated when the Get request for an alarm variable returns an error. The specific error is identified by jnxRmonAlarmGetFailReason.
jnxRmonGetOk	jnxRmonTrapPrefix 2	Generated when the Get request for an alarm variable is successful. This trap is only sent after previous attempts are unsuccessful.

Chapter 25

Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB

The reverse-path-forwarding MIB monitors statistics for traffic that is rejected because of reverse-path-forwarding (RPF) processing. The reverse-path-forwarding MIB includes one main object, jnxRpfStats, with an object identifier of {jnxRpf 1}. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-rpf.txt.

This chapter discusses the following topic:

jnxRpfStatsTable

The jnxRpfStatsTable, whose object identifier is {jnxRpfStats 1}, provides a list of RPF entries in table format.

jnxRpfStatsEntry

The jnxRpfStatsEntry, whose object identifier is {jnxRpfStatsTable 1}, has four objects, which are listed in Table 69.

Table 69: jnxRpfStatsEntry

Object	Object Identifier	Description
jnxRpfStatsIfIndex	jnxRpfStatsEntry 1	The ingress interface for traffic that is counted in an RpfStats entry.
jnxRpfStatsAddrFamily	jnxRpfStatsEntry 2	The address family of an entry's traffic, which can be in IPv4 or IPv6 format
jnxRpfStatsPackets	jnxRpfStatsEntry 3	The number of packets received on this interface, belonging to this address family, that have been rejected due to RPF processing.
jnxRpfStatsBytes	jnxRpfStatsEntry 4	The number of bytes received on this interface, belonging to this address family, that have been rejected due to RPF processing.

Chapter 26

Interpreting the Enterprise-Specific Source Class Usage MIB

The source class usage (SCU) MIB counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge.

The SCU MIB is an object of the jnxMibs branch of the enterprise-specific MIB {enterprise 2636} and has an object identifier of {jnxMIB 16}. The SCU MIB includes one object, jnxScuStats, which has an object identifier of {jnxScu 1}. For information about configuring source and destination class usage, see the *JUNOS Policy Framework Configuration Guide* and the *JUNOS Network Interfaces Configuration Guide*. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-scu.txt.

This chapter discusses the following topic:

jnxScuStatsTable

The entries in the jnxScuStatsTable, whose object identifier is {jnxScuStats 1}, are represented by jnxScuStatsEntry, whose object identifier is {jnxScuStatsTable 1}, and are listed in Table 70.

Table 70: jnxScuStatsEntry

Object	Object Identifier	Description
jnxScuStatsDstIfIndex	jnxScuStatsEntry 1	The destination interface index, which is the egress interface of traffic that is counted by this table entry.
jnxScuStatsAddrFamily	jnxScuStatsEntry 2	The address family of an entry's traffic in IPv4 or IPv6 format.
jnxScuStatsClassName	jnxScuStatsEntry 3	The name of the source class. All traffic counted in this table entry satisfies the requirements defined by this source class.

Object	Object Identifier	Description
jnxScuStatsPackets	jnxScuStatsEntry 4	The number of packets sent out of jnxScuStatsDstIfIndex that match the source class (jnxScuStatsClassName) and the address type (jnxScuStatsAddrFamily) defined for a table entry.
jnxScuStatsBytes	jnxScuStatsEntry 5	The number of bytes sent out of jnxScuStatsDstIfIndex that match the source class (jnxScuStatsClassName) and the address type (jnxScuStatsAddrFamily) defined for a table entry.
jnxScuStatsClName	jnxScuStatsEntry 6	The name of the source class. This object is a duplicate of jnxScuStatsClassName and is included to satisfy those network management applications that cannot extract the class name from the instance portion of the OID.

Chapter 27

Interpreting the Enterprise-Specific Passive Monitoring MIB

The passive monitoring MIB, whose object identifier is {jnxMibs 19}, performs traffic flow monitoring and lawful interception of packets transiting between two routers. This MIB allows you to do the following:

- Gather and export detailed information about Internet Protocol version 4 (IPv4) traffic flows between source and destination nodes in your network.
- Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.
- The passive monitoring MIB has three tables: jnxPMonFlowTable, JnxPMonErrorTable, and jnxPMonMemoryTable. jnxPMonFlowTable monitors and collects statistics on the flow of traffic on a Passive Monitoring Physical Interface Card (PIC). jnxPMonErrorTable monitors and collects statistics on packet and memory errors on a Passive Monitoring PIC. jnxPMonMemoryTable monitors and collects statistics on memory usage on a Passive Monitoring PIC. For information about system requirements, see the *JUNOS Feature Guide*. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-pmon.txt.

This chapter documents only jnxPMonFlowTable.

This chapter contains the following topic:

jnxPMonFlowTable

jnxPMonFlowTable has an object identifier of {jnxPMon 1}. Its entries are represented by JnxPMonFlowEntry, which contains the objects listed in Table 71.

Table 71: jnxPMFlowEntry

Object	Object Identifier	Description
jnxPMonCurrentActiveFlows	jnxPMonFlowEntry 1	Monitors the number of currently active flows on a Passive Monitoring PIC.
jnxPMonTotalFlows	jnxPMonFlowEntry 2	Monitors the total flows on a Passive Monitoring PIC.
jnxPMonTotalFlowsPackets	jnxPMonFlowEntry 3	Monitors the total packet flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowsPackes	jnxPMonFlowEntry 4	Monitors the number of packets in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsBytes	jnxPMonFlowEntry 5	Monitors the number of total of bytes in all flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowBytes	jnxPMonFlowEntry 6	Monitors the number of bytes in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsExpired	jnxPMonFlowEntry 7	Monitors the number of total flows expired on a Passive Monitoring PIC.
jnxPMonTotalFlowsAged	jnxPMonFlowEntry 8	Monitors the number of total flows aged on a Passive Monitoring PIC.
jnxPMonTotalFlowsExported	jnxPMonFlowEntry 9	Monitors the number of total flows exported on a Passive Monitoring PIC.
jnxPMonTotalFlowsPacketsExported	jnxPMonFlowEntry 10	Monitors the number of total flow packets exported on a Passive Monitoring PIC.

Chapter 28

Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB

The SONET/SDH interface management MIB sends the current alarm state for each SONET/SDH interface. When the alarm state changes on an interface, the MIB updates its alarm status. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sonet.txt.

This chapter discusses the following topic:

jnxSonetAlarmsTable

The jnxSonetAlarmsTable, whose object identifier is {jnxSonetAlarm 1}, provides information about alarm status on SONET/SDH physical interfaces.

jnxSonetAlarmEntry

The jnxSonetAlarmEntry, whose object identifier is {jnxSonetAlarmTable 1}, has five objects, which are listed in table 72.

Table 72: jnxSonetAlarmTable

Object	Object Identifier	Description
jnxSonetCurrentAlarms	jnxSonetAlarmEntry 1	Identifies all the active SONET/SDH alarms on this interface.
jnxSonetLastAlarmId	jnxSonetAlarmEntry 2	Identifies the SONET/SDH alarm that most recently was set or cleared.
jnxSonetLastAlarmTime	jnxSonetAlarmEntry 3	The value of sysUpTime when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmDate	jnxSonetAlarmEntry 4	The system date and time when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmEvent	jnxSonetAlarmEntry 5	Indicates whether the last alarm event set a new alarm or cleared an existing alarm.

Table 73 provides an example of jnxSonetAlarmInterface objects on an M20 router.

Table 73: jnxSonetAlarmInterface Objects in the jnxSonetAlarmTable of an M20 router

Alarm Interface	CurrentAlarms	Last Alarm ID	Last Alarm Time (System Up Time)	Last Alarm Date and Time	Last Alarm Event
14	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.15	2002-10-15, 10:21:14.0,-7:0	set(2)
15	sonetLosAlarm(3)	sonetLosAlarm(3)	8 days, 4:09:46.22	2002-10-23,14:29:23.0,-7:0	set(2)
16	sonetLolAlarm(0) sonetLosAlarm(3)	sonetBerrSdAlarm(8)	8 days, 4:09:46.21	2002-10-23,14:29:23.0,-7:0	cleared(3)
17	sonetLofAlarm(2)	sonetLaisAlarm(5)	8 days, 4:09:47.21	2002-10-23,14:29:24.0,-7:0	cleared(3)
18	————————	sonetLosAlarm(3)	7 days, 4:31:27.53	2002-10-22,14:51:4.0,-7:0	cleared(3)
19	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.16	2002-10-15,10:21:14.0,-7:0	set(2)
20	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.17	2002-10-15,10:21:14.0,-7:0	set(2)
21	————————	sonetLofAlarm(2)	7 days, 11:15:00.15	2002-10-22,21:34:37.0,-7:0	cleared(3)
22	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLolAlarm(0)	7 days, 6:33:32.02	2002-10-22,16:53:8.0,-7:0	set(2)
23	————————	sonetLosAlarm(3)	7 days, 6:33:45.02	2002-10-22,16:53:21.0,-7:0	cleared(3)
24	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.07	2002-10-15,10:21:14.0,-7:0	set(2)
25	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.08	2002-10-15,10:21:14.0,-7:0	set(2)
26	————————	————————	0:00:00.00	0-0-0,0:0:0.0,	none(1)
27	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
28	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
29	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)

Chapter 29

Interpreting the Enterprise-Specific SONET APS MIB

The SONET Automatic Protection Switching (APS) management MIB monitors any SONET interface that participates in APS. APS is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The JUNOS implementation of APS allows you to protect against circuit failures between an ADM and one or more routers, and between multiple interfaces in the same router. When a circuit or router fails, a backup immediately takes over. For more information about APS, see the *JUNOS Network Interfaces Configuration Guide*.

NOTE: The JUNOS software supports only read access, 1 + 1 architecture, bidirectional, revertive, and nonrevertive mode.

For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sonetaps.txt.

This chapter discusses the following topics:

apsConfigTable

apsConfigTable lists the APS groups that are configured on the system.

apsConfigEntry

apsConfigEntry objects have read access only and are listed in Table 74.

Table 74: apsConfigTable

Object	Object Identifier	Description
apsConfigName	apsConfigEntry 1	A text name for the APS group. An entry cannot exist in the active state unless all objects in the entry have an appropriate value. Also, all associated apsChanConfigEntry rows must represent a set of consecutive channel numbers beginning with 0 or 1, depending on the selected architecture.
apsConfigRowStatus	apsConfigEntry 2	The status of a APS group entry.
apsConfigMode	apsConfigEntry 3	The architecture of the APS group. The JUNOS software supports only the 1 + 1 architecture.
apsConfigRevert	apsConfigEntry 4	The revertive mode of the APS group. ■ Revertive mode—When the condition that caused a switch to the protection line has been cleared, the signal is switched back to the working line. Switching can optionally be revertive with 1 + 1 architecture. ■ Nonrevertive mode—Traffic remains on the protection line until another switch request is received.
apsConfigDirection	apsConfigEntry 5	The directional mode of the APS group. The JUNOS software supports only bidirectional mode. Bidirectional mode provides protection in both directions.
apsConfigExtraTraffic	apsConfigEntry 6	This object always returns the value disabled.
apsConfigSdBerThreshold	apsConfigEntry 7	The signal degrade bit error rate (BER). The negative value of this number is used as the exponent of 10 for computing the threshold value for the BER. For example, a value of 5 indicates a BER threshold of 10^-5.
apsConfigSfBerThreshold	apsConfigEntry 8	The signal failure bit error rate. The negative value of this number is used as the exponent of 10 for computing the threshold value for the BER. For example, a value of 5 indicates a BER threshold of 10^-5.

Object	Object Identifier	Description
apsConfigWaitToRestore	apsConfigEntry 9	The wait to restore period, in seconds. After a condition that necessitated an automatic switch is cleared, the wait to restore period must elapse before reverting. This avoids rapid switch oscillations. GR-253-CORE specifies a range of 5 to 12 minutes. G.783 defines a 5 to 12 minute range in section 5.4.1.1.3, but also allows a shorter period in Table 2-1, WaitToRestore value (MI_WTRtime: 0..(5)..12 minutes).
apsConfigCreationTime	apsConfigEntry 10	The value of sysUpTime at the time the row was created
apsConfigStorageType	apsConfigEntry 11	The storage type for this conceptual row. For information about conceptual rows, see RFC 2579, *Textual Conventions for SMIv2*.

apsStatusTable

apsStatustable provides status information about configured APS groups.

apsStatusEntry

apsStatusEntry objects have read access only and are listed in Table 75.

Table 75: apsStatusTable

Object	Object Identifier	Description
apsStatusK1K2Rcv	apsStatusEntry 1	The current value of the K1 and K2 bytes received on the protection channel.
apsStatusK1K2Trans	apsStatusEntry 2	The current value of the K1 and K2 bytes transmitted on the protection channel.
apsStatusCurrent	apsStatusEntry 3	The current status of the APS group. This object has the following values: ■ modeMismatch—Modes other than 1 + 1 unidirectional monitor protection line K2 bit 5, which indicates the architecture, and K2 bits 6 through 8, which indicate if the mode is unidirectional or bidirectional. A conflict between the current local mode and the received K2 mode information constitutes a mode mismatch. The JUNOS software supports only bidirectional mode. ■ channelMismatch—A mismatch between the transmitted K1 channel and the received K2 channel has been detected.

Object	Object Identifier	Description
apsStatusCurrent (cont.)	apsStatusEntry 3	■ psbf—A protection switch byte failure (PSBF) is in effect. This condition occurs when either an inconsistent APS byte or an invalid code is detected. An inconsistent APS byte occurs when no 3 consecutive K1 bytes of the last 12 successive frames are identical, starting with the last frame containing a previously consistent byte. An invalid code occurs when the incoming K1 byte contains an unused code or a code irrelevant for the specific switching operation (for example, reverse request while no switching request is outstanding) in three consecutive frames. An invalid code also occurs when the incoming K1 byte contains an invalid channel number in three consecutive frames. ■ feplf—Modes other than 1 + 1 unidirectional monitor the K1 byte for far-end protection-line failures. A far-end protection-line defect is declared based on receiving a signal failure (SF) on the protection line. ■ extraTraffic—Indicates whether extra traffic is currently being accepted on the protection line. ■ extraTraffic—Indicates whether extra traffic is currently being accepted on the protection line.
apsStatusModeMismatches	apsStatusEntry 4	Counts mode mismatch conditions. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime.
apsStatusChannelMis-matches	apsStatusEntry 5	Counts channel mismatch conditions. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime.

Object	Object Identifier	Description
apsStatusPSBFs	apsStatusEntry 6	Counts protection switch byte failure conditions. This condition occurs when either an inconsistent APS byte or an invalid code is detected. An inconsistent APS byte occurs when no 3 consecutive K1 bytes of the last 12 successive frames are identical, starting with the last frame containing a previously consistent byte. An invalid code occurs when the incoming K1 byte contains an unused code or a code irrelevant for the specific switching operation (for example, reverse request while no switching request is outstanding) in three consecutive frames. An invalid code also occurs when the incoming K1 byte contains an invalid channel number in three consecutive frames. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime.
apsStatusFEPLFs	apsStatusEntry 7	Counts far-end protection-line failure conditions. This condition is declared based on receiving a signal failure (SF) on the protection line in the K1 byte. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime.
apsStatusSwitchedChannel	apsStatusEntry 8	This field is set to the number of the channel that is currently switched to protection. The value 0 indicates that no channel is switched to protection. The values 1 through 14 indicate that the working channel is switched to protection.
apsStatusDiscontinuity-Time	apsStatusEntry 9	The value of sysUpTime when the last one or more of this APS group's counters experienced a discontinuity. The relevant counters are the specific instances associated with this APS group of any Counter32 object contained in apsStatusTable. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value.

apsChanConfigTable

apsChanConfigTable lists the APS channels that have been configured in APS groups.

apsChanConfigEntry

apsChanConfigEntry objects have read access only and are listed in Table 76.

Table 76: apsChanConfigTable

Object	Object Identifier	Description
apsChanConfigGroupName	apsChanConfigEntry 1	A text name for the APS group in which this channel is included.
apsChanConfigNumber	apsChanConfigEntry 2	A unique channel number within an APS group. The value 0 indicates the null channel. The values 1 through 14 define a working channel.
apsChanConfigRowStatus	apsChanConfigEntry 3	The status of this APS channel entry. An entry cannot exist in the active state unless all objects in the entry have an appropriate value. The JUNOS software supports only 1 + 1 architecture. The values 1 through 14 define a working channel. When an attempt is made to set the corresponding apsConfigRowStatus field to active, the apsChanConfigNumber values of all entries with equal apsChanConfigGroupName fields must be a set of consecutive integer values beginning with 0 or 1, depending on the architecture of the group, and ending with *n*, where *n* is greater than or equal to 1 and less than or equal to 14. Otherwise, the error inconsistentValue is returned to the apsConfigRowStatus set attempt.
apsChanConfigIfIndex	apsChanConfigEntry 4	The interface index assigned to a SONET LTE. This is an interface with ifType sonet(39). The value of this object must be unique among all instances of apsChanConfigIfIndex. In other words, a particular SONET LTE can only be configured in one APS group. This object cannot be set if the apsChanConfigGroupName instance associated with this row is equal to an instance of apsConfigName and the corresponding apsConfigRowStatus object is set to active. In other words, this value cannot be changed if the APS group is active. However, this value can be changed if the apsConfigRowStatus value is equal to notInService. The JUNOS software supports only read access.

Object	Object Identifier	Description
apsChanConfigPriority	apsChanConfigEntry 5	The priority of the channel. This field returns the value low priority. The JUNOS software supports only 1 + 1 architecture.
apsChanConfigStorageType	apsChanConfigEntry 6	The storage type for this conceptual row. Conceptual rows having the value permanent need not allow write access to any columnar objects in the row. For information about conceptual rows, see RFC 2579, *Textual Conventions for SMIv2*.

apsChanStatusTable

apasChanStatusTable provides APS channel statistics.

apsChanStatusEntry

apsChanConfigEntry objects have read access only and are listed in Table 77.

Table 77: apsChanStatusTable

Object	Object Identifier	Description
apsChanStatusCurrent	apsChanStatusEntry 1	The current state of the port. This object has the following values: lockedOut—This bit, when applied to a working channel, indicates that the channel is prevented from switching to the protection line. When applied to the null channel, this bit indicates that no working channel can switch to the protection line. sd—A signal degrade condition is in effect. sf—A signal failure condition is in effect switched. The switched bit is applied to a working channel if that channel is currently switched to the protection line. wtr—A wait-to-restore state is in effect.
apsChanStatusSignalDegrades	apsChanStatusEntry 2	A count of signal degrade conditions. A signal degrade condition occurs when the line bit error rate exceeds the currently configured value of the relevant instance of apsConfigSdBerThreshold. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.
apsChanStatusSignalFailures	apsChanStatusEntry 3	A count of signal failure conditions that have been detected on the incoming signal. A signal failure condition occurs when a loss of signal, loss of frame, AIS-L or line bit error rate exceeds the currently configured value of the relevant instance of apsConfigSfBerThreshold. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.

Object	Object Identifier	Description
apsChanStatusSwitchovers	apsChanStatusEntry 4	When queried with index value apsChanConfigNumber other than 0, this object returns the number of times this channel has switched to the protection line. When queried with index value s set to 0, which is the protection line, this object returns the number of times that any working channel has switched back to the working line from this protection line. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.
apsChanStatusLastSwitchover	apsChanStatusEntry 5	When queried with index value apsChanConfigNumber other than 0, this object returns the value of sysUpTime when this channel last completed a switch to the protection line. If this channel has never switched to the protection line, the value 0 is returned. When queried with index value apsChanConfigNumber set to 0, which is the protection line, this object will return the value of sysUpTime the last time that a working channel was switched back to the working line from this protection line. If no working channel has ever switched back to the working line from this protection line, the value 0 is returned.

Object	Object Identifier	Description
apsChanStatusSwitchoverSeconds	apsChanStatusEntry 6	The cumulative Protection Switching Duration (PSD) time, in seconds. For a working channel, this is the cumulative number of seconds that service was carried on the protection line. For the protection line, this is the cumulative number of seconds that the protection line has been used to carry any working channel traffic. This information is only valid if revertive switching is enabled. The value 0 will be returned. Otherwise, discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of apsChanStatusDiscontinuityTime. For example, if the value of an instance of apsChanStatusSwitchoverSeconds changes from a non-zero value to zero due to revertive switching being disabled. It is expected that the corresponding value of apsChanStatusDiscontinuityTime is updated to reflect the time of the configuration change.
apsChanStatusDiscontinuityTime	apsChanStatusEntry 7	The value of sysUpTime on the most recent occasion at which any one or more of this channel's counters suffered a discontinuity. The relevant counters are the specific instances associated with this channel of any Counter32 object contained in apsChanStatusTable. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value for apsChanStatusEntry.

Chapter 30

Interpreting the Enterprise-Specific Ethernet MAC MIB

The Ethernet media access control (MAC) MIB, whose object identifier is {jnxMibs 23}, monitors media access control statistics on Gigabit Ethernet intelligent queuing (IQ) interfaces. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-mac.txt.

This chapter discusses the following topic:

jnxMacStatsTable

The jnxMacStatsTable contains a list of MAC statistics for Gigabit Ethernet interfaces.

jnxMacStatsEntry

jnxMacStatsEntry has six objects, which are listed in Table 78.

Table 78: jnxMacStatsTable

Object	Object Identifier	Description
jnxVlanIndex	jnxMacStatsEntry 1	The virtual LAN (VLAN) ID of a VLAN.
jnxSourceMacAddress	jnxMacStatsEntry 2	The source MAC address.
jnxMacHCInOctets	jnxMacStatsEntry 3	The number of total octets received in this VLAN/MAC address.
jnxMacHCInFrames	jnxMacStatsEntry 4	The number of total frames received in this VLAN/MAC address
jnxMacHCOutOctets	jnxMacStatsEntry 5	The number of total octets transmitted in this VLAN/MAC address.
jnxMacHCOutFrames	jnxMacStatsEntry 6	The number of total frames transmitted in this VLAN/MAC address.

Chapter 31

Interpreting the Enterprise-Specific Interface MIB

The interface MIB extends the standard ifTable (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-if-extensions.txt.

This chapter discusses the following topics:

jnxIfTable

jnxIfTable lists traffic statistics, input errors, and output errors for an interface.

jnxIfEntry

jnxIfEntry objects are listed in Table 79.

Table 79: jnxIfTable

Object	Object Identifier	Description
ifIn1SecRates	ifJnxEntry 1	The number of bits per second delivered by this sublayer to its next higher sublayer.
ifIn1SecOctets	ifJnxEntry 2	The number of octets per second delivered by this sublayer to its next higher sublayer.
ifIn1SecPkts	ifJnxEntry 3	The number of packets per second delivered by this sublayer to its next higher sublayer.
ifOut1SecRate	ifJnxEntry 4	The number of bits per second delivered by this sublayer to its next lower sublayer.
ifOut1SecOctets	ifJnxEntry 5	The number of octets per second delivered by this sublayer to its next lower sublayer.
ifOut1SecPkts	ifJnxEntry 6	The number of packets per second delivered by this sublayer to its next lower sublayer.
ifHCIn1SecRate	ifJnxEntry 7	The number of bits per second delivered by this sublayer to its next higher sublayer. This object is a 64-bit version of ifIn1SecRate.
ifHCOut1SecRate	ifJnxEntry 8	The number of bits per second delivered by this sublayer to its next lower sublayers. This object is a 64-bit version of ifOut1SecRate.
ifJnxInErrors	ifJnxEntry 9	Errors: The sum of the incoming frame aborts and FCS errors.
ifJnxInFrameErrors	ifJnxEntry 10	Framing Errors: The number of input packets that were misaligned.
ifJnxInQDrops	ifJnxEntry 11	Drops: The number of packets dropped by the input queue of the I/O Manager ASIC.
ifJnxInRunts	ifJnxEntry 12	Runts: Frames received that are smaller than the runt threshold.
ifJnxInGiants	ifJnxEntry 13	Giants: Frames received that are larger than the giant threshold.
ifJnxInDiscards	ifJnxEntry 14	Policed discards: Frames that the incoming packet match code discarded because they were not recognized or of interest.
ifJnxInHsICrcErrors	ifJnxEntry 15	HS link CRC errors: The number of CRC errors on the high-speed links between the ASICs responsible for handling the router interfaces while receiving packets.
ifJnxInHsIFifoOverFlows	ifJnxEntry 16	HS link FIFO overflows: The number of FIFO overflows on the high-speed links between the ASICs responsible for handling the router interfaces.

Object	Object Identifier	Description
ifJnxInL3Incompletes	ifJnxEntry 17	L3 incompletes: The number of incoming packets that fail Layer 3 sanity checks of the header.
ifJnxInL2ChanErrors	ifJnxEntry 18	L2 channel errors: The number of incoming packets for which the software could not find a valid logical interface.
ifJnxInL2MismatchTimeouts	ifJnxEntry 19	L2 mismatch timeouts: The count of malformed or short packets that cause the incoming packet handler to discard the frame as unreadable.
ifJnxInInvalidVCs	ifJnxEntry 20	Invalid VCs: The number of cells that arrived for a nonexistent virtual circuit
ifJnxInFifoErrors	ifJnxEntry 21	FIFO errors: The number of FIFO errors in the received direction as reported by the ASIC on the PIC.
ifJnxBucketDrops	ifJnxEntry 22	Bucket drops: Drops because traffic load exceeded the interface transmit and receive leaky bucket configuration.
ifJnxSramErrors	ifJnxEntry 23	SRAM errors: This counter increments when a hardware error has occurred in the SRAM on the PIC.
ifJnxOutErrors	ifJnxEntry 24	Errors: The sum of the outgoing frame aborts and FCS errors.
ifJnxCollisions	ifJnxEntry 25	Collisions: The number of output collisions detected on this interface.
ifJnxCarrierTrans	ifJnxEntry 26	Carrier transitions: The number of times the interface saw the carrier signal transition.
ifJnxOutQDrops	ifJnxEntry 27	Drops: The number of packets dropped by the output queue of the I/O Manager ASIC.
ifJnxOutAgedErrors	ifJnxEntry 28	Aged packets: The number of packets that remained in shared packet SDRAM for so long that the system automatically purged them.
ifJnxOutFifoErrors	ifJnxEntry 29	FIFO errors: The number of FIFO errors in the transmit direction as reported by the ASIC on the PIC.
ifJnxOutHslFifoUnderFlows	ifJnxEntry 30	HS link FIFO underflows: The number of FIFO underflows on the high-speed links between the ASICs responsible for handling the router interfaces.
ifJnxOutHslCrcErrors	ifJnxEntry 31	HS link CRC errors: The number of CRC errors on the high-speed links between the ASICs responsible for handling the router interfaces while transmitting packets.

ifChassisTable

ifChassisTable provides additional interface and chassis information.

ifChassisEntry

ifChassisEntry objects are listed in Table 80.

Table 80: ifChassisTable

Object	Object Identifier	Description
ifChassisFpc	ifChassisEntry 1	The number of the FPC card on which the interface is located in the chassis. It is the chassis slot in which the FPC card is installed for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to Network Management convention. Therefore, a value of zero means there is no real or physical FPC associated with the specified interface.
ifChassisPic	ifChassisEntry 2	The number of the PIC card on which the interface is located in the chassis. It is the PIC location on the FPC card for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to Network Management convention. Therefore, a value of zero means there is no real or physical PIC associated with the specified interface.
ifChassisPort	ifChassisEntry 3	The number of the port on the PIC card on which the interface is located in the chassis. It is the port number on the PIC card for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to Network Management convention. Therefore, a value of zero means there is no real or physical port associated with the specified interface.
ifChassisChannel	ifChassisEntry 4	The channel identifier for the specified interface if it is part of a channelized interface. Although the channel is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to Network Management convention. For an interface that could not be channelized, this object returns zero.

Object	Object Identifier	Description
ifChassisLogicalUnit	ifChassisEntry 5	The logical unit number of the specified interface. It is the logical part of the interface that is configured on the physical or channel part, if any. Although the logical unit number is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to Network Management convention. For an interface that is really a physical device, this value returns zero.
ifChassisPicIndex	ifChassisEntry 6	The indexes for the chassis MIB tables. This is the instance index that keys into jnxContentsTable in the chassis MIB. For example, the octet string of 8.1.2.0 means a PIC ("8" first digit) at FPC slot 0 ("1-1", second digit minus one if nonzero) PIC number 1 ("2-1", third digit) minus one if nonzero port number whatever (fourth digit currently unused) which in turn could be plugged in by NMS directly after any MIB objects in the jnxContentsTable gets that PIC object for the specified interface. This object is valid only for interfaces having real and physical PIC cards. Otherwise, it returns an octet string "0.0.0.0."

Chapter 32

Interpreting the Enterprise-Specific VPN MIB

The enterprise-specific virtual private network (VPN) MIB, whose object identifier is {jnxMibs 26}, provides monitoring for the following type of VPNs:

- Layer 2 based on Internet draft draft-kompella-l2ppvpn-version.txt, *MPLS-based Layer 2 VPNs*.
- Layer 3 based on Internet draft draft-ietf-l3vpn-rfc2547bis-03.txt, *BGP and MPLS IP VPNs*.
- VPLS based on Internet draft draft-ietf-ppvpn-vpls-bgp-00.txt, *Virtual Private LAN Service*.

NOTE: The Simple Network Management Protocol (SNMP) cannot be associated with any routing instances other than the master routing instance.

For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/ junos74/swconfig74-net-mgmt/html/mib-jnx-vpn.txt.

This chapter discusses the following topics:

jnxVpnInfo

jnxVpnInfo, whose object identifier is jnxVpnMibObjects 1, contains information about the number of configured VPNs and active VPNs.

Table 81 lists the supported jnxVpnInfo objects, VPNs, and circuit connection services.

Table 81: Supported jnxVpnInfo Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	Circuit Cross-Connect	L2 Circuit	Optical VPN	Description
jnxVpnConfiguredVpns	jnxVpnInfo 1	Yes	Yes	Yes	No	Yes	NA	Number of configured VPNs.
jnxVpnActiveVpns	jnxVpnInfo 2	Yes	Yes	Yes	No	Yes	NA	Number of active VPNs.
jnxVpnNextIfIndex	jnxVpnInfo 3	NA	NA	NA	NA	NA	NA	Next free VPN interface index.
jnxVpnNextPwIndex	jnxVpnInfo 4	NA	NA	NA	NA	NA	NA	Next free pseudo-wire index.
jnxVpnNextRTIndex	jnxVpnInfo 5	NA	NA	NA	NA	NA	NA	Next free route target index.

jnxVpnTable

jnxVpnTable, whose object identifier is jnxVpnMibobjects 2, lists configured VPNs.

jnxVpnEntry

JnxVpnEntry contains information about a configured VPN with the objects listed in Table 82 and their supported VPNs and circuit connection services. The first two objects in jnxVpnEntry (JnxVpnType and JnxVpnname) are indexes and are not included in this table.

Table 82: Supported jnxVpnEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRowStatus	jnxVpnEntry 3	NA	NA	NA	NA	NA	NA	Creates, modifies, or deletes a row in this table.
JnxVpnStorageType	jnxVpnEntry 4	NA	NA	NA	NA	NA	NA	The storage type.
jnxVpnDescription	jnxVpnEntry 5	Yes	Yes	Yes	Yes	No	NA	VPN description.
jnxVpnIdentifierType	jnxVpnEntry 6	Yes	Yes	Yes	Yes	No	NA	Type of jnxVpnIdentifer.
jnxVpnIdentifier	jnxVpnEntry 7	Yes	Yes	Yes	Yes	No	NA	For Border Gateway Protocol (BGP) VPNs, the route distinguisher for the VPN. For Label Distribution Protocol (LDP) VPNs, the virtual circuit (VC) ID for the circuit. A value of all zeros indicates that a route distinguisher and a VC ID are not configured for the VPN.
jnxVpnConfiguredSites	jnxVpnEntry 8	NA	No	No	No	No	NA	The number of sites configured in the VPN.
jnxVpnActiveSites	jnxVpnEntry 9	NA	No	No	No	No	NA	The number of active sites in the VPN.
jnxVpnLocalAddresses	jnxVpnEntry 10	No	No	No	No	No	NA	The number of addresses learned from the CE device.
jnxVpnTotalAddresses	jnxVpnEntry 11	No	No	No	No	No	NA	The total number of addresses in the VPN routing table.
jnxVpnVpnAge	jnxVpnEntry 12	Yes	Yes	Yes	Yes	No	NA	How old the VPN is, in hundredths of a second.

jnxVpnIfTable

The jnxVpnIfTable, whose object identifier is jnxVpnMibObjects 3, lists VPN interfaces.

jnxVpnIfEntry

jnxVpnIfEntry contains information about VPN interfaces, and has the objects listed in Table 83. The first three objects (jnxVpnIfVpnType, jnxVpnIfVpnName, and jnxVpnIfIndex) are indexes and are not included in this table.

Table 83: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfRowStatus	jnxVpnifEntry 4	NA	NA	NA	NA	NA	NA	Creates, modifies, or deletes a row in this table.
jnxVpnIfStorageType	jnxVpnifEntry 5	NA	NA	NA	NA	NA	NA	Identifies the storage type for an object.
jnxVpnIfAssociationPw	jnxVpnifEntry 6	NA	Yes	Yes	Yes	No	NA	The index of the associated pseudo-wire. If there no index is associated with a pseudo-wire, the index is 0. A pseudo-wire is a mechanism that carries essential elements of an emulated circuit from one provider edge (PE) device to one or more other PEs over a PSN.

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfProtocol	jnxVpnifEntry 7	No	Yes	Yes	Yes	No	NA	Indicates the protocol running over a VPN interface. This object contains the following values: ■ other(0) ■ frameRelay(1) ■ atmAal5(2) ■ atmCell(3) ■ ethernetVlan(4) ■ ethernet(5) ■ ciscoHdlc(6) ■ ppp(7) ■ cem(8) ■ atmVcc(9) ■ atmVpc(10) ■ vpls(11) ■ ipInterworking(12) ■ snapInterworking(13) ■ static(20) ■ rip(21) ■ ospf(22) ■ bgp(23) ■ atmTrunkNNI (129) ■ atmTrunkUNI (130)
jnxVpnIfInBandwidth	jnxVpnifEntry 8	No	No	No	No	No	NA	The maximum bandwidth that the customer edge (CE) device connected over a VPN can send to the PE device, in kilobytes per second. A value of "0" indicates that there is no configured maximum.

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfOutBandwidth	jnxVpnifEntry 9	No	No	No	No	No	NA	The maximum bandwidth that the PE device can send to the CE device over a VPN interface, in kilobytes per second. A value of "0" indicates that there is no configured maximum.
jnxVpnIfStatus	jnxVpnifEntry 10	Yes	Yes	Yes	Yes	No	NA	Displays the status of a monitored VPN interface. This object contains the following values: ■ unknown(0) ■ noLocalInterface(1) ■ disabled(2) ■ encapsulation-Mismatch(3) ■ down(4) ■ up(5)

jnxVpnPwTable

jnxVpnPwTable, whose object identifier is jnxVpnMibObjects 4, lists pseudo-wire connections.

jnxVpnPwEntry

jnxVpnPwEntry contains pseudo-wire information about a VPN that is being monitored, and has the objects listed in Table 84. The first three objects (jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex) are indexes and are not listed in this table.

Table 84: Supported jnxVpnEntry Objects, VPNs, and Connection Circuit Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwRowStatus	jnxVpnPwEntry 4	NA	NA	NA	NA	NA	NA	Creates, modifies, and deletes a row in this table.
jnxVpnPwStorageType	jnxVpnPwEntry 5	NA	NA	NA	NA	NA	NA	The storage type.
jnxVpnPwAssociatedInterface	jnxVpnPwEntry 6	NA	Yes	Yes	Yes	No	NA	The VPN index of the interface associated with a pseudo-wire. If there is no interface associated with a pseudo-wire, 0 is returned.
jnxVpnPwLocalSiteId	jnxVpnPwEntry 7	NA	Yes	Yes	Yes	No	NA	The local site identifier for a pseudo-wire. When there is no local site identifier, 0 is returned.
jnxVpnPwRemoteSiteId	jnxVpnPwEntry 8	NA	Yes	Yes	Yes	No	NA	The remote site identifier. For example, the site at the end of the pseudo-wire. When there is no remote site identifier, 0 is returned.
jnxVpnRemotetPeIdAddrType	jnxVpnPwEntry 9	NA	Yes	Yes	Yes	No	NA	The remote PE address. For example, the router at the end of the pseudo-wire.

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRemotePeIdAddress	jnxVpnPwEntry 10	NA	Yes	Yes	Yes	No	NA	The type of the tunnel over which the pseudo-wire is carried. If several pseudo-wires can be carried in one tunnel, each pseudo-wire is identified by the multiplexer or demultiplexer within a tunnel. This object can contain the following values: ■ static(1) ■ gre(2) ■ l2tpv3(3) ■ ipSec(4) ■ ldp(5) ■ rsvpTe(6) ■ crLdp(7)
jnxVpnPwTunnelType	jnxVpnPwEntry 11	NA	Yes	Yes	Yes	No	NA	The type of tunnel over which the pseudo-wire is carried.
jnxVpnPwTunnelName	jnxVpnPwEntry 12	NA	Yes	Yes	Yes	No	NA	The name of the tunnel over which a pseudo-wire is carried.
jnxVpnPwReceiveDemux	jnxVpnPwEntry 13	NA	Yes	Yes	Yes	No	NA	The demultiplexer value that identifies received the packets associated with this pseudo-wire.
jnxVpnPwTransmitDemux	jnxVpnPwEntry 14	NA	Yes	Yes	Yes	No	NA	The demultiplexer value that identifies the transmitted packets associated with this pseudo-wire.

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwStatus	jnxVpnPwEntry 15	NA	Yes	Yes	Yes	No	NA	The status of the pseudo-wire. This object can have the following values: ■ unknown(0) ■ down(1) ■ up(2)
jnxVpnPwTunnelStatus	jnxVpnPwEntry 16	NA	No	No	No	No	NA	The status of the PE-to-PE tunnel over which the pseudo-wire is carried.
jnxVpnPwRemoteSiteStatus	jnxVpnPwEntry 17	NA	No	No	No	No	NA	The interface status at the remote end of the pseudo-wire. This object can have the following values: ■ unknown(0) ■ outOfRange(1) ■ down(2) ■ up(3)
jnxVpnPwTimeUp	jnxVpnPwEntry 18	NA	Yes	Yes	Yes	No	NA	The time, in hundredths of a second, that a pseudo-wire has been operational.
jnxVpnPwTransitions	jnxVpnPwEntry 19	NA	Yes	Yes	Yes	No	NA	The number of state transitions (up to down and down to up) that a tunnel has undergone.
jnxVpnPwLastTransition	jnxVpnPwEntry 20	NA	Yes	Yes	Yes	No	NA	The time, in hundredths of a second, since the last transition occurred in a tunnel.
jnxVpnPwPacketsSent	jnxVpnPwEntry 21	NA	No	No	No	No	NA	The number of packets sent over a pseudo-wire.
jnxVpnPwOctetsSent	jnxVpnPwEntry 22	NA	No	No	No	No	NA	The number of octets sent over a pseudo-wire.

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwPacketsReceived	jnxVpnPwEntry 23	No	No		No	No	NA	The number of packets received over a pseudo-wire.
jnxVpnPwOctetsReceived	jnxVpnPwEntry 24	No	No		No	No	NA	The number of octets received over a pseudo-wire.
jnxVpnPwLRPacketsSent	jnxVpnPwEntry 25	No	No		No	No	NA	The number of packets sent over a pseudo-wire.
jnxVpnPwLROctetsSent	jnxVpnPwEntry 26	No	No		No	No	NA	The number of octets sent over a pseudo-wire.
jnxVpnPwLRPacketsReceived	jnxVpnPwEntry 27	No	No		No	No	NA	The number of packets received over a pseudo-wire.
jnxVpnPwLROctetsReceived	jnxVpnPwEntry 28	No	No		No	No	NA	The number of octets received over a pseudo-wire.

jnxVpnRTTable

The jnxVpnRTTable, whose object identifier is jnxVpnMibObjects 4, contains route targets for a VPN.

jnxVpnRTEntry

jnxVpnRTEntry lists route targets for a given VPN, and has the objects listed in Table 85. The first three objects (jnxVpnRTVpnType, jnxVpnRTVpnName, and jnxVpnRTIndex) are indexes and are not listed in this table.

Table 85: Supported jnxVpnRTEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRTRowStatus	jnxVpnRTEntry 4	NA	NA	NA	NA	NA	NA	Creates, modifies, or deletes a row in this table.
jnxVpnRTStorageType	jnxVpnRTEntry 5	NA	NA	NA	NA	NA	NA	Identifies the storage type for an object.
jnxVpnRTType	jnxVpnRTEntry 6	Yes	Yes	Yes	NA	No	NA	The type of the following route target. The type can be routeTarget[012] or none.
jnxVpnRT	jnxVpnRTEntry 7	Yes	Yes	Yes	NA	No	NA	The VPN route target. If jnxVpnRTType is none, the value must be all zeros.
jnxVpnRTFunction	jnxVpnRTEntry 8	Yes	Yes	Yes	NA	No	NA	The route target export distribution type.

VPN Traps

The enterprise-specific VPN MIB provides traps for monitoring VPNs. Table 86 lists supported VPN traps, VPNs, and circuit connection services.

Table 86: Supported VPN Traps, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfUp	jnxVpnMIBnotificatios 1	Yes	Yes	NA	Yes	No	NA	Indicates that the interface with the index jnxVpnIfIndex belonging to the jnxVpnIfVpnName of type jnxVpnIfVpnType went up.
jnxVpnIfDown	jnxVpnMIBnotificatios 2	Yes	Yes	NA	Yes	No	NA	Indicates that the interface with index jnxVpnIfIndex belonging to jnxVpnIfVpnName of type jnxVpnIfVpnType went down.
jnxVpnPwUp	jnxVpnMIBnotificatios 3	Yes	Yes	Yes	Yes	No	NA	Indicates that the pseudo-wire with the index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went up.
jnxVpnPwDown	jnxVpnMIBnotificatios 4	Yes	Yes	Yes	Yes	No	NA	Indicates that the pseudo-wire with index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went down.

Chapter 33

Interpreting the Enterprise-Specific Flow Collection Services MIB

The enterprise-specific flow collection services MIB, whose object identifier is {jnxMibs 28}, provides statistics on files, records, memory, FTP, and error states of flow collection services on a Monitoring Services Physical Interface Card (PIC). It also provides Simple Network Management (SNMP) traps for unavailable destinations, unsuccessful file transfers, flow overloading, and memory overloading. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-coll.txt. For information about how to configure the flow collection services interface, see the *JUNOS Services Interfaces Configuration Guide* and the *JUNOS Feature Guide*.

This chapter discusses the following topics:

jnxCollGlobalStats

jnxCollGlobalStats provides statistics on all the router's Monitoring Services PICs and has the objects listed in Table 87.

Table 87: jnxCollGlobalStats

Object	Object Identifier	Description
jnxCollGlobalCreatedFiles	jnxCollGlobalStats 1	The number of files created by all the Monitoring Services PICs on the router since the last time the router was reset.
jnxCollGlobalOpenFiles	jnxCollGlobalStats 2	The number of open flow collection services files currently on the router.

jnxCollPicIfTable

jnxCollPicIfTable contains statistics about each Monitoring Services PIC.

jnxCollPicEntry

JnxCollPicEntry contains information about a Monitoring Services PIC. Each Monitoring Services PIC contains one interface and is identified by IfIndex. It has objects listed in Table 88.

Table 88: jnxCollPicEntry

Object	Object Identifier	Description
File Statistics		
jnxCollPicIfCreatedFiles	jnxCollPicIfEntry 1	The number of files created by a Monitoring Services PIC since the last time the PIC was reset.
jnxCollPicIfCreatedFileRate	jnxCollPicIfEntry 2	The number of files created per second during the current 10-second interval.
jnxCollPicIfPeakCreatedFileRate	jnxCollPicIfEntry 3	The peak number of files created per second.
jnxCollPicIfExportedFiles	jnxCollPicIfEntry 4s	The number of files exported by a Monitoring Services PIC.
jnxCollPicIfExportedFileRate	jnxCollPicIfEntry 5	The number of files exported per second during the current 10-second interval.
jnxCollPicIfPeakExportedFileRate	jnxCollPicIfEntry 6	The peak number of files exported per second.
jnxCollPicIfDestroyedFiles	jnxCollPicIfEntry 7	The number of files successfully exported and files dropped by the Monitoring Services PIC. Files are destroyed after they are transferred to the FTP server or when there is not enough memory.
jnxCollPicIfDestroyedFileRate	jnxCollPicIfEntry 8	The number of files dropped per second during the current 10-second interval. Files are dropped after they are transferred to the FTP server or when there is not enough memory.
jnxCollPicIfPeakDestroyedFileRate	jnxCollPicIfEntry 9	The peak number of files dropped, per second. Files are dropped after they are transfered to the FTP server or when there is not enough memory.

Object	Object Identifier	Description
Record Statistics		
jnxCollPicIfProcRecords	jnxCollPicIfEntry 10	The number of flow records processed by a Monitoring Services PIC.
jnxCollPicIfProcRecordsRate	jnxCollPicIfEntry 11	The number of flow records processed per seconds during the current 10-second interval.
jnxCollPicIfPeakProcRecordsRate	jnxCollPicIfEntry 12	The peak number of flow records processed.
Memory Statistics		
jnxCollPicIfMemoryUsed	jnxCollPicIfEntry 13	The amount of memory used, in bytes, by a Monitoring Services PIC.
jnxCollPicIfMemoryFree	jnxCollPicIfEntry 14	The amount of memory free, in bytes, by a Monitoring Services PIC.
FTP Statistics		
jnxCollPicIfFtpBytes	jnxCollPicIfEntry 15	The number of bytes transferred via FTP by a Monitoring Services PIC.
jnxCollPicIfFtpByteRate	jnxCollPicIfEntry 16	The number of bytes per second transfered via FTP measured during the current 10-second interval.
jnxCollPicIfPeakFtpByteRate	jnxCollPicIfEntry 17	The peak number of bytes per second transferred via FTP.
jnxCollPicIfFtpFiles	jnxCollPicIfEntry 18	The number of files transferred via FTP by a Monitoring Services PIC.
jnxCollPicIfFtpFileRate	jnxCollPicIfEntry 19	The number of files per second transferred via FTP.
jnxCollPicIfPeakFtpFileRate	jnxCollPicIfEntry 20	The peak number of files per second transferred via FTP.
jnxCollPicIfFtpFailures	jnxCollPicIfEntry 21	The number of FTP transfer failures transferred by a Monitoring Services PIC.
Error State Statistics		
jnxCollPicIfCurrentState	jnxCollPicIfEntry 22	The current state of various error conditions on a Monitoring Services PIC.
jnxCollPicIfLastStateChange	jnxCollPicIfEntry 23	The error condition of the last changed state.
jnxCollPicIfStateChangeTime	jnxCollPicIfEntry 24	The value of sysUpTime when the management subsystem last learned of a change to the jnxCollPicIfCurrentState for a Monitoring Services PIC.
jnxCollPicIfStateChangeDate	jnxCollPicIfEntry 25	The system date and time when the management subsystem last learned of a change to the jnxCollPicIfCurrentState on a Monitoring Services PIC.
jnxCollPicIfStateChangeType	jnxCollPicIfEntry 26	Indicates whether the last state change set a new error condition or cleared an existing one. This object contains the following values: ■ none(1) ■ set(2) ■ cleared(3)

jnxCollFileTable

jnxCollFileTable contains information about each flow collection services file on the router.

jnxCollFileEntry

jnxCollFileEntry contains information about a single file open on a Monitoring Services PIC, and has the objects listed in Table 89.

Table 89: jnxCollFileTable

Object	Object Identifier	Description
jnxCollFileName	jnxCollFileEntry 1	The name of a flow collection services file on a Monitoring Services PIC.
jnxCollFileFname	jnxCollFileEntry 2	The name of a flow collection services file on this Monitoring Services PIC. This object is included for those Network Management (NM) applications that can't parse the filename from the instance portion of the OIDs and provides the value of jnxCollFileName.
jnxCollFileRecords	jnxCollFileEntry 3	The number of flow records in this file.
jnxCollFileRecordRate	jnxCollFileEntry 4	The number of flow records per second added to this file, measured during the current 10-second interval.
jnxCollFilePeakRecordRate	jnxCollFileEntry 5	The peak number of flow records per second added to this file.
jnxCollFileUncompBytes	jnxCollFileEntry 6	The number of uncompressed bytes in this file.
jnxCollFileUncompByteRate	jnxCollFileEntry 7	The number of uncompressed bytes per second added to this file.
jnxCollFilePeakUncompByteRate	jnxCollFileEntry 8	The peak number of uncompressed bytes per second added to this file.
jnxCollFileCompBytes	jnxCollFileEntry 9	The number of compressed bytes in this file.
jnxCollFileCompByteRate	jnxCollFileEntry 10	The number of compressed bytes per second added to this file during the current 10-second interval.
jnxCollFilePeakCompByteRate	jnxCollFileEntry 11	The peak number of compressed bytes per second added to this file.
jnxCollFileBlocks	jnxCollFileEntry 12	The number of blocks in this file.
jnxCollFileCompBlocks	jnxCollFileEntry 14	The number of compressed blocks in this file.
jnxCollFileTransferAttempts	jnxCollFileEntry 15	The number of FTP transfer attempts in this file.
jnxCollFileState	jnxCollFileEntry 16	The current state of this file. This object contains the following values: ■ unknown(1) ■ active(2)—The file is actively receiving flow records. ■ wait(3)—The file is waiting for export. ■ export1(4)—The file is being exported to the primary server. ■ export2(5)—The file is being exported to the secondary server.

Chapter 34

Interpreting the Enterprise-Specific Services PIC MIB

The Adaptive Services (AS) Physical Interface Card (PIC) allows you to provide multiple services on a single PIC by configuring a set of services and applications. The AS PIC offers a special range of services you configure in one or more service sets: stateful firewalls, Network Address Translation (NAT), and intrusion detection services (IDS).

The Services PIC MIB, whose object identifier is {jnxMibs 32}, sends the current operational status for each Adaptive Services PIC. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos74/swconfig74-net-mgmt/html/mib-jnx-sp.txt.

This chapter discusses the following topics:

NOTE: The Services PIC MIB is not supported on J-series Services Routers unless the appropriate services license is enabled.

jnxSpSvcSetTable

The jnxSpSvcSetTable, whose object identifier is {jnxSPSvcSet 1}, provides information about each service set on each Adaptive Services PIC on the router.

jnxSpSvcSetEntry

The jnxSpSvcSetEntry, whose object identifier is {jnxSpSvcSetTable 1}, has 11 objects, which are listed in Table 90 on page 390. Each entry provides information about a single service set. The service set is identified by the name of the service set. The Adaptive Services PIC on which the service set is configured is identified by JnxSpSvcSetIFName.

Table 90: jnxSpSvcSetTable

Object	Object Identifier	Description
jnxSpSvcSetName	jnxSpSvcSetEntry 1	A text name for the service set.
jnxSpSvcSetSvcType	jnxSpSvcSetEntry 2	The name of the service type associated with the service set.
jnxSpSvcSetTypeIndex	jnxSpSvcSetEntry 3	An integer used to identify the service type for the service set.
jnxSpSvcSetIfName	jnxSpSvcSetEntry 4	The name of the interface identifying the Adaptive Services PIC. If more than one interface is associated with the Adaptive Services PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetIfIndex	jnxSpSvcSetEntry 5	An index number associated with the interface name.
jnxSpSvcSetMemoryUsage	jnxSpSvcSetEntry 6	Amount of memory used by the service set, in bytes.
jnxSpSvcSetCpuUtil	jnxSpSvcSetEntry 7	Amount of CPU processing used by the service set, expressed as a percentage of total CPU usage. J-series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.
jnxSpSvcSetSvcStyle	jnxSpSvcSetEntry 8	Type of service for the service set. Service types include: Unknown —The service type is not known. Interface-service — The service is interface based. Next-hop-service — The service is next-hop based.
jnxSpSvcSetMemLimitPktDrops	jnxSpSvcSetEntry 9	Number of packets dropped because the service set exceeded its memory limits (operating in the Red zone).
jnxSpSvcSetCpuLimitPktDrops	jnxSpSvcSetEntry 10	Number of packets dropped because the service set exceeded the average CPU limits (when total CPU usage exceeds 85 percent).
jnxSpSvcSetFlowLimitPktDrops	jnxSpSvcSetEntry 11	Number of packets dropped because the service set exceeded the flow limit.

jnxSpSvcSetSvcTypeTable

The jnxSpSvcSetSvcTypeTable, whose object identifier is {jnxSPSvcSet 2}, provides information about each service on each Adaptive Services PIC on the router. The stateful firewall, NAT, or IDS service sets are categorized as one SvcType (SFW/NAT/IDS).

jnxSpSvcSetSvcTypeEntry

The jnxSpSvcSetSvcTypeEntry, whose object identifier is {jnxSpSvcSetSvcTypeTable 1}, has seven objects, which are listed in Table 91. Each entry provides information about a single service on each Adaptive Services PIC. Each Adaptive Services PIC is identified by its corresponding index number, while each service is identified by jnxSpSvcSetSvcTypeIndex. The service type associated with this index is provided by jnxSpSvcSetSvcTypeName.

Table 91: jnxSpSvcSetSvcTypeTable

Object	Object Identifier	Description
jnxSpSvcSetSvcTypeIndex	jnxSpSvcSetSvcTypeEntry 1	An integer used to identify the service type.
jnxSpSvcSetSvcTypeIfName	jnxSpSvcSetSvcTypeEntry 2	The name of the interface identifying the Adaptive Services PIC. If more than one interface is associated with the Adaptive Services PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetSvcTypeName	jnxSpSvcSetSvcTypeEntry 3	The name of the service type.
jnxSpSvcSetSvcTypeSvcSets	jnxSpSvcSetSvcTypeEntry 4	Number of service sets configured on the Adaptive Services PIC that use this service type.
jnxSpSvcSetSvcTypeMemoryUsage	jnxSpSvcSetSvcTypeEntry 5	Amount of memory used by this service type, expressed in bytes.
jnxSpSvcSetSvcTypePctMemoryUsage	jnxSpSvcSetSvcTypeEntry 6	Amount of memory used by this service type, expressed as a percentage of total memory.
jnxSpSvcSetSvcTypeCpuUtil	jnxSpSvcSetSvcTypeEntry 7	Amount of CPU processing used by the service set, expressed as a percentage of total CPU usage. J-series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.

jnxSpSvcSetIfTable

The jnxSpSvcSetIfTable, whose object identifier is {jnxSPSvcSet 3}, provides service set information for each Adaptive Serices PIC on the router.

jnxSpSvcSetSvcIfEntry

The jnxSpSvcSetIfEntry, whose object identifier is {jnxSpSvcSetIfTable 1}, has eight objects, which are listed in Table 92. Each entry provides service set information about a single Adaptive Services PIC. Each Adaptive Services PIC is identified by its corresponding index number.

Table 92: jnxSpSvcSetIfTable

Object	Object Identifier	Description
jnxSpSvcSetIfTableName	jnxSpSvcSetIfEntry 1	The name of the interface used to identify the Adaptive Services PIC. If more than one interface is associated with the Adaptive Services PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetIfsvcSets	jnxSpSvcSetIfEntry 2	The number of service sets configured on the Adaptive Services PIC.
jnxSpSvcSetIfMemoryUsage	jnxSpSvcSetIfEntry 3	Amount of memory used by the Adaptive Services PIC, expressed in bytes.
jnxSpSvcSetIfPctMemoryUsage	jnxSpSvcSetIfEntry 4	Amount of memory used by the Adaptive Services PIC, expressed as a percentage of total memory.
jnxSpSvcSetIfPolMemoryUsage	jnxSpSvcSetIfEntry 5	Amount of policy memory used by the Adaptive Services PIC, expressed in bytes.
jnxSpSvcSetIfPctPolMemoryUsage	jnxSpSvcSetIfEntry 6	Amount of policy memory used by the Adaptive Services PIC, expressed as a percentage of the total.
jnxSpSvcSetIfMemoryZone	jnxSpSvcSetIfEntry 7	The memory usage zone currently occupied by the Adaptive Services PIC. The definitions of each zone are: ■ Green—All new flows are allowed. ■ Yellow—Unused memory is reclaimed. All new flows are allowed. ■ Orange—New flows are allowed only for service sets that use less than their equal share of memory. ■ Red—No new flows are allowed.
jnxSpSvcSetIfCpuUtil	jnxSpSvcSetIfEntry 8	Amount of CPU processing used by the Adaptive Services PIC, expressed as a percentage of total CPU usage. J-series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.

Service Traps

The enterprise-specific Services PIC MIB provides traps for monitoring AS PICs. Table 93 lists the supported traps.

Table 93: Supported Traps for Services PIC MIB

Object	Object Identifier	Description
jnxSpSvcSetZoneEntered	jnxSPNotificationPrefix 1	Indicates that an Adaptive Services PIC has entered a more severe memory usage zone from a less severe memory usage zone. The zone entered is identified by JnxSpSvcSetIfMemoryZone.
jnxSpSvcSetZoneExited	jnxSPNotificationPrefix 2	Indicates that an Adaptive Services PIC has exited a more severe memory usage zone to a less severe memory usage zone. The zone entered is identified by JnxSpSvcSetIfMemoryZone.
jnxSpSvcSetCpuExceeded	jnxSPNotificationPrefix 3	Indicates that an Adaptive Services PIC has over 85 percent CPU usage. This trap is not supported on J-series Services Routers.
jnxSpSvcSetCpuOk	jnxSPNotificationPrefix 4	Indicates that an Adaptive Services PIC has returned to less than 85 percent CPU usage. This trap is not supported on J-series Services Routers.

Redundant Interfaces

On M-series and T-series platforms, redundant adaptive services interfaces (rsp) appear in the jnxSpSvcSetIfTable just like any other adaptive services interface (sp). With the exception of the index, information presented for an rsp interface is similar to the underlying sp interface. In the jnxSpSvcSetTable, only the underlying sp interface is shown because the Adaptive Services PIC does not track the overlying rsp interface,

Part 7
Accounting Options

Chapter 35

Accounting Options Overview

An accounting profile represents common characteristics of collected accounting data, including the following:

- Collection interval
- File to contain accounting data
- Specific fields and counter names on which to collect statistics

You can configure multiple accounting profiles, as described in Table 94.

Table 94: Types of Accounting Profiles

Type of Profile	Description
Interface profile	Collects the specified error and statistic information.
Filter profile	Collects the byte and packet counts for the counter names specified in the filter profile.
Routing Engine profile	Collects selected Routing Engine statistics and logs them to a specified file.
Class usage profile	Collects class usage statistics and logs them to a specified file.

Chapter 36

Configuring Accounting Options

To configure accounting options, include the following statements at the [edit accounting-options] hierarchy level of the configuration:

```
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        destination-classes {
            destination-class-name;
        }
        source-classes {
            source-class-name;
    }
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
    }
    filter-profile profile-name {
        counters {
            counter-name;
        }
        file filename;
        interval minutes;
        }
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
```

```
        routing-engine-profile profile-name {
            fields {
                field-name;
            }
            file filename;
            interval minutes;
        }
    }
```

By default, accounting options are disabled.

This section describes the minimum required configuration and discusses the following tasks for configuring accounting options:

Minimum Accounting Options Configuration

To enable accounting options on the router, you must perform at least the following tasks:

- Configure accounting options by including a **file** statement and one or more **source-class-usage**, **destination-class-profile**, **filter-profile**, **interface-profile**, or **routing-engine-profile** statements at the **[edit accounting-options]** hierarchy level:

```
[edit]
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        source-classes {
            source-class-name;
        destination-classes {
            destination-class-name;
        }
    }
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
    }
    filter-profile profile-name {
        counters {
            counter-name;
        }
        file filename;
        interval minutes;
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
    routing-engine-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
}
```

- Apply the profiles to the chosen interfaces or filters.

Apply an interface profile to a physical or logical interface by including the accounting-profile statement at either the [edit interfaces *interface-name*] or the [edit interfaces *interface-name* unit *number*] hierarchy level. For more information on interface profiles, see the *JUNOS Network Interfaces Configuration Guide*.

```
[edit interfaces]
interface-name {
    accounting-profile profile-name;
    unit number {
        accounting-profile profile-name;
    }
}
```

NOTE: You do not apply destination class profiles to interfaces. Although the interface needs to have the destination-class-usage statement configured, the destination class profile automatically finds all interfaces with the destination class configured.

Apply a filter profile to a firewall filter by including the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level:

```
[edit firewall]
filter filter-name {
    accounting-profile profile-name;
}
```

You do not need to apply the Routing Engine profile to an interface because the statistics are collected on the Routing Engine itself.

Configuring Files

An accounting profile specifies what statistics should be collected and written to a log file. To configure an accounting-data log file, include the file statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
        }
    }
```

filename is name of file in which to write accounting data.

If the filename contains spaces, enclose it in quotation marks (" "). The filename cannot contain a forward slash (/). The file is created in the /var/log directory and can contain data from multiple profiles.

All accounting-data log files include header and trailer sections that start with a # in the first column. The header contains the file creation time, the hostname, and the columns that appear in the file. The trailer contains the time that the file was closed.

Whenever any configured value changes that affects the columns in a file, the file creates a new profile layout record that contains a new list of columns.

You must configure the file size; all other properties are optional.

- Configuring the Maximum Size of the File on page 404
- Configuring the Maximum Number of Files on page 404
- Configuring the Transfer Interval of the File on page 404
- Configuring Archive Sites on page 404

Configuring the Maximum Size of the File

To configure the maximum size of the files, include the size statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
size bytes;
```

The size statement is the maximum size of the log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). The minimum value for *bytes* is 256 KB. You must configure *bytes*; the remaining attributes are optional.

Configuring the Maximum Number of Files

To configure the maximum number of the files, include the files statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
files filenumber;
```

The files statement specifies the maximum number of files. When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. The minimum value for *filenumber* is 3 and the default value is 10.

Configuring the Transfer Interval of the File

To configure the transfer interval of the files, include the transfer-interval statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
transfer-interval minutes;
```

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Configuring Archive Sites

After a file reaches its maximum size or the transfer-interval time is exceeded, the file is closed, renamed, and, if you configured an archive site, transferred to a remote host. To configure archive sites, include the archive-sites statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
archive-sites {
    site-name;
}
```

site-name is any valid FTP URL. For more information on how to specify valid FTP URLs, see the *JUNOS System Basics Configuration Guide*. You can specify more than one URL, in any order. When a file is archived, the router attempts to transfer the file to the first URL in the list, trying the next site in the list only if the transfer does not succeed. The log file is stored at the archive site with a filename of the format *router-name_log-filename_timestamp*.

Configuring the Interface Profile

An interface profile specifies the information collected and written to a log file. You can configure a profile to collect error and statistic information for input and output packets on a particular physical or logical interface.

To configure an interface profile, include the interface-profile statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
interface-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

Each accounting profile must have a unique *profile-name*. To apply a profile to a physical or logical interface, include the accounting-profile statement at either the [edit interfaces *interface-name*] or the [edit interfaces *interface-name* unit *number*] hierarchy level. You can also apply a accounting profile at the [edit firewall family *family-type* filter *filter-name*] hierarchy level. For more information, see the *JUNOS Policy Framework Configuration Guide*.

To configure an interface profile, you perform the tasks described in the following sections:

- Configuring Fields on page 405
- Configuring the File Information on page 406
- Configuring the Interval on page 406
- Example: Configuring the Interface Profile on page 406

Configuring Fields

An interface profile must specify what statistics are collected. To configure which statistics should be collected for an interface, include the fields statement at the [edit accounting options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
fields {
    field-name;
}
```

Configuring the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
file filename;
```

You must specify a filename statement for the interface profile that has already been configured at the [edit accounting-options] hierarchy level.

Configuring the Interval

Each interface with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
interval minutes;
```

NOTE: The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of interfaces might cause serious performance degradation.

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configuring the Interface Profile

Configure the interface profile:

```
[edit]
accounting-options {
    file if_stats {
        size 40 files 5;
    }
    interface-profile if_profile1 {
        file if_stats;
        interval 30;
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-multicast;
            output-multicast;
        }
    }
    interface-profile if_profile2 {
        file if_stats;
        interval 30;
        fields {
```

```
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-multicast;
            output-multicast;
        }
    }
    interfaces {
        ge-1/0/0 {
            accounting-profile if_profile1;
            unit 0 {
                accounting-profile if_profile2;
            ...
            }
        }
    }
```

The two interface profiles, if-profile1 and if-profile2, write data to the same file, if-stats. The if-stats file might look like the following:

```
#FILE CREATED 976823478 2000-12-14-19:51:18
#hostname host
#profile-layout
if_profile2,epoch-timestamp,interface-name,snmp-index,input-bytes,output-bytes,
input-packets,output-packets,input-multicast,output-multicast
#profile-layout
if_profile1,epoch-timestamp,interface-name,snmp-index,input-bytes,output-bytes,
input-packets
if_profile2,976823538,ge-1/0/0.0,8,134696815,3681534,501088,40723,0,0
if_profile1,976823538,ge-1/0/0,7,134696815,3681534,501088
...
#FILE CLOSED 976824378 2000-12-14-20:06:18
```

Configuring the Filter Profile

A filter profile specifies error and statistics information collected and written to a file. A filter profile must specify counter names for which statistics are collected. To configure a filter profile, include the filter-profile statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
filter-profile profile-name {
    counters {
        counter-name;
    }
    file filename;
    interval minutes;
}
```

To apply the filter profile, include the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level. For more information on firewall filters, see the *JUNOS Network Interfaces Configuration Guide*.

To configure a filter profile, perform the tasks described in the following sections:

Configuring the Counters

Statistics are collected for all counters specified in the filter profile. To configure the counters, include the counters statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
counters {
    counter-name;
}
```

Configuring the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
file filename;
```

You must specify a filename for the filter profile that has already been configured at the [edit accounting options] hierarchy level.

NOTE: If the configured file size or transfer interval is exceeded, the JUNOS software closes the file and starts a new one. By default, the transfer interval value is 30 minutes. If the transfer interval is not configured, the JUNOS software closes the file and starts a new one when the file size exceeds its configured value or the default transfer interval value exceeds 30 minutes. To avoid transferring files every 30 minutes, specify a different value for the transfer interval.

Configuring the Interval

Each filter with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
interval minutes;
```

NOTE: The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of filters might cause serious performance degradation.

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configuring a Filter Profile

Configure a filter profile:

```
[edit]
accounting-options {
    file fw_accounting {
        size 500k files 4;
    }
    filter-profile fw_profile1 {
        file fw_accounting;
        interval 60;
        counters {
            counter1;
            counter2;
            counter3;
        }
    }
}
firewall {
    filter myfilter {
        accounting-profile fw_profile1;
        ...
        term accept-all {
        then {
            count counter1;
            accept;
        }
    }
}
```

The filter profile, fw-profile1, writes data to the file fw_accounting. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#hostname host
#profile-layout
fw_profile1,epoch-timestamp,filter-name,counter-name,packet-count,byte-count
fw_profile1,976826058,myfilter,counter1,163,10764
...
#FILE CLOSED 976826178 2000-12-14-20:36:18
```

Example: Configuring Interface-Specific Firewall Counters and Filter Profiles

To collect and log count statistics collected by firewall filters on a per-interface basis, you must configure a filter profile and include the interface-specific statement at the [edit firewall filter *filter-name*] hierarchy level.

Configure the firewall filter accounting profile:

```
[edit accounting-options]
    file cust1_accounting {
    size 500k;
}
filter-profile cust1_profile {
    file cust1_accounting;
    interval 1;
    counters {
      r1;
    }
}
```

Configure the interface-specific firewall counter:

```
[edit firewall]
filter f3 {
    accounting-profile cust1_profile;
    interface-specific;
    term f3-term {
      then {
        count r1;
          accept;
      }
    }
}
```

Apply the firewall filter to an interface:

```
[edit interfaces]
ge-1/0/0 {
    unit 0 {
      family inet {
        filter {
          input f3;
          output f3;
        }
        address 20.20.20.30/24;
      }
    }
}
```

The following example shows the contents of the cust1_accounting file in the /var/log folder that might result from the preceding configuration:

```
#FILE CREATED 995495212 2001-07-18-22:26:52
#hostname host
#profile-layout cust1_profile,epoch-timestamp,interfaces,filter-name,
counter-name,packet-count,byte-count
cust1_profile,995495572,ge-1/0/0.0,f3-ge-1/0/0.0-i,r1-ge-1/0/0.0-i,5953,1008257
cust1_profile,995495602,ge-1/0/0.0,f3-ge-1/0/0.0-o,r1-ge-1/0/0.0-o,5929,1006481
...
```

If the interface-specific statement is not included in the configuration, the following output might result:

```
#FILE CREATED 995495212 2001-07-18-22:26:52
#hostname host
#profile-layout cust1_profile,epoch-timestamp,interfaces,filter-name,
counter-name,packet-count,byte-count
cust1_profile,995495572,ge-1/0/0.0,f3,r1,5953,1008257
cust1_profile,995495632,ge-1/0/0.0,f3,r1,5929,1006481
```

Configuring Source Class Usage Options

You can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as *source classes* and *destination classes*. You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces.

Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

On T-series and M320 routing platforms, the source class and destination classes are not carried across the platform fabric. The implications of this are as follows:

- On T-series and M320 platforms, SCU and DCU accounting is performed before the packet enters the fabric.
- On T-series and M320 routing platforms, DCU is performed before output filters are evaluated. On M-series platforms, DCU is performed after output filters are evaluated.
- If an output filter drops traffic on M-series platforms, the dropped packets are excluded from DCU statistics. If an output filter drops traffic on T-series and M320 routing platforms, the dropped packets are included in DCU statistics.

For more information about source class usage, see the *JUNOS Policy Framework Configuration Guide*, the *JUNOS Network Interfaces Configuration Guide,* and the *JUNOS Feature Guide.*

To configure source class usage options, perform the tasks described in this section:

- Configuring SCU and/or DCU
- Configuring SCU on a Virtual Loopback Interface on page 414
- Configuring Class Usage Profiles on page 415

Configuring SCU and/or DCU

To configure SCU and/or DCU, perform the following tasks described in this section:

- Creating Prefix Route Filters in a Policy Statement on page 412
- Applying the Policy to the Forwarding Table on page 412
- Enabling Accounting on Inbound and Outbound Interfaces on page 413

Creating Prefix Route Filters in a Policy Statement

Define prefix router filters:

```
[edit policy-options]
policy-statement scu-1 {
      term term1;
         from {
            route-filter 192.168.1.0/24 orlonger;
         }
         then source-class gold;
}
```

Applying the Policy to the Forwarding Table

Apply the policy to the forwarding table:

```
[edit]
routing-options {
      forwarding-table {
            export scu-1;
         }
}
```

Enabling Accounting on Inbound and Outbound Interfaces

You can enable accounting inbound and outbound interfaces:

```
[edit]
interfaces {
    so-6/1/0 {
        unit 0 {
            family inet;
                accounting {
                    destination-class-usage;
                    source-class-usage {
                        output;
                    }
                }
            }
        }
    }
}
[edit]
interfaces {
    ge-0/1/0 {
        unit 0 {
            family inet6 {
                accounting {
                    source-class-usage {
                        input;
                    }
                }
            }
        }
    }
}
```

Optionally, you can include the input and output statements on a single interface:

```
[edit]
interfaces {
    ge-0/1/2 {
        unit 0 {
            family inet6 {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
            }
        }
    }
}
```

For more information on configuring route filters and source classes in a routing policy, see the *JUNOS Policy Framework Configuration Guide* and the *JUNOS Network Interfaces Configuration Guide*.

Configuring SCU on a Virtual Loopback Interface

To configure source class usage on the virtual loopback interface, perform the tasks described in the following sections:

- Example: Configuring a Virtual Loopback Interface on a Provider Edge Router Equipped with a Tunnel PIC on page 414
- Example: Mapping the VRF Instance Type to the Virtual Loopback Interface on page 414
- Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface on page 415

Example: Configuring a Virtual Loopback Interface on a Provider Edge Router Equipped with a Tunnel PIC

Define a virtual loop interface on a provider edge router with a Tunnel PIC:

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

Example: Mapping the VRF Instance Type to the Virtual Loopback Interface

Map the VRF instance type to the virtual loopback interface:

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface at-2/1/1.0;
        interface vt-0/3/0.0;
        route-distinguisher 10.255.14.225:100;
        vrf-import import-policy-name;
        vrf-export export-policy-name;
        protocols {
            bgp {
                group to-r4 {
                    local-address 10.27.253.1;
                    peer-as 400;
                    neighbor 10.27.253.2;
                }
            }
        }
    }
}
```

NOTE: For SCU and DCU to work, do not include the vrf-table-label statement at the [edit routing-instances *instance-name*] hierarchy level.

Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface

Send traffic received from the virtual loopback interface out of the source class output interface:

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

For more information on configuring source class usage on the virtual loopback interface, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring Class Usage Profiles

To collect class usage statistics, perform the tasks described in these sections:

- Configuring a Class Usage Profile on page 415
- Configuring the File Information on page 416
- Configuring the Interval on page 416
- Creating a Class Usage Profile to Collect Source Class Usage Statistics on page 416
- Creating a Class Usage Profile to Collect Destination Class Usage Statistics on page 417

Configuring a Class Usage Profile

You can configure the class usage profile to collect statistics for particular source and destination classes.

To configure the class usage profile to filter by source classes, include the source-classes statement at the [edit accounting options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
source-classes {
    source-class-name;
}
```

To configure the class usage profile to filter by destination classes, include the destination-classes statement at the [edit accounting options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
destination-classes {
    destination-class-name;
}
```

Configuring the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting-options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
file filename;
```

You must specify a filename for the source class usage profile that has already been configured at the [edit accounting options] hierarchy level. You can also specify a filename for the destination class usage profile configured at the [edit accounting options] hierarchy level.

Configuring the Interval

Each interface with a class usage profile enabled has statistics collected once per interval specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
interval minutes;
```

Creating a Class Usage Profile to Collect Source Class Usage Statistics

To create a class usage profile to collect source class usage statistics:

```
[edit]
accounting-options {
    class-usage-profile scu-profile1;
    file usage-stats;
    interval 15;
    source-classes {
      gold;
      silver;
      bronze;
    }
}
```

The class usage profile, scu-profile1, writes data to the file usage_stats. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, scu_profile,epoch-timestamp,interface-name,source-class,
packet-count,byte-count
scu_profile,980313078,ge-1/0/0.0,gold,82,6888
scu_profile,980313078,ge-1/0/0.0,silver,164,13776
scu_profile,980313078,ge-1/0/0.0,bronze,0,0
scu_profile,980313678,ge-1/0/0.0,gold,82,6888
scu_profile,980313678,ge-1/0/0.0,silver,246,20664
scu_profile,980313678,ge-1/0/0.0,bronze,0,0
```

Creating a Class Usage Profile to Collect Destination Class Usage Statistics

To create a class usage profile to collect destination class usage statistics:

```
[edit]
accounting-options {
    class-usage-profile dcu-profile1;
    file usage-stats
    interval 15;
    destination-classes {
      gold;
      silver;
      bronze;
    }
}
```

The class usage profile, dcu-profile1, writes data to the file usage-stats. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, dcu_profile,epoch-timestamp,interface-name,destination-class,
packet-count,byte-count
dcu_profile,980313078,ge-1/0/0.0,gold,82,6888
dcu_profile,980313078,ge-1/0/0.0,silver,164,13776
dcu_profile,980313078,ge-1/0/0.0,bronze,0,0
dcu_profile,980313678,ge-1/0/0.0,gold,82,6888
dcu_profile,980313678,ge-1/0/0.0,silver,246,20664
dcu_profile,980313678,ge-1/0/0.0,bronze,0,0
...
#FILE CLOSED 976826178 2000-12-14-20:36:18
```

Configuring the Routing Engine Profile

The Routing Engine profile collects Routing Engine statistics and logs them to a file. The Routing Engine profile specifies the fields for which statistics are collected.

To configure a Routing Engine profile, include the routing-engine-profile statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
routing-engine-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

To configure a Routing Engine profile, perform the tasks described in the following sections:

- Configuring Fields on page 418
- Configuring the File Information on page 418
- Configuring the Interval on page 419
- Example: Configuring a Routing Engine Profile on page 419

Configuring Fields

A Routing Engine profile must specify what statistics are collected. To configure which statistics should be collected for the Routing Engine, include the fields statement at the [edit accounting options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
fields {
    field-name;
}
```

Configuring the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
file filename;
```

You must specify a *filename* for the Routing Engine profile that has already been configured at the [edit accounting-options] hierarchy level.

Configuring the Interval

A Routing Engine profile has statistics collected once per interval time specified for the profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
interval minutes;
```

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configuring a Routing Engine Profile

Configure a Routing Engine profile:

```
[edit accounting-options]
file my-file {
    size 300k;
}
routing-engine-profile profile-1 {
    file my-file;
    fields {
        host-name;
        date;
        time-of-day;
        uptime;
        cpu-load-1;
        cpu-load-5;
        cpu-load-15;
    }
}
```

Chapter 37

Summary of Accounting Options Configuration Statements

The following sections explain each of the accounting options configuration statements. The statements are organized alphabetically.

accounting-options

Syntax accounting-options {...}

Hierarchy Level [edit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure options for accounting statistics collection.

Usage Guidelines See "Configuring Accounting Options" on page 399.

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

archive-sites

Syntax
```
archive-sites {
    site-name;
}
```

Hierarchy Level [edit accounting-options file *filename*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure an archive site. If more than one site name is configured, an ordered list of archive sites for the accounting-data log files is created. When a file is archived, the router 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 a filename of the format *router-name_log-filename_timestamp*.

Options *site-name*—Any valid FTP URL to a destination. For information on how to specify valid FTP URLs, see the *JUNOS System Basics Configuration Guide*.

Usage Guidelines See “Configuring Archive Sites” on page 404.

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

class-usage-profile

Syntax

```
class-usage-profile profile-name {
    file filename;
    interval minutes;
    source-classes {
      source-class-name;
    }
    destination-classes {
      destination-class-name;
    }
}
```

Hierarchy Level [edit accounting-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a class usage profile, which is used to log class usage statistics to a file in the /var/log directory. The class usage profile logs class usage statistics for the configured source classes on every interface that has destination-class-usage configured.

For information on configuring source classes, see the *JUNOS Routing Protocols Configuration Guide*. For information on configuring source class usage, see the *JUNOS Network Interfaces Configuration Guide*.

Options *profile-name*—Name of the destination class profile.

The remaining statements are explained separately.

Usage Guidelines See “Configuring Class Usage Profiles” on page 415.

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

counters

Syntax
```
counters {
    counter-name;
}
```

Hierarchy Level [edit accounting-options filter-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Names of counters for which filter profile statistics are collected. The packet and byte counts for the counters are logged to a file in the /var/log directory.

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

Usage Guidelines See "Configuring the Counters" on page 408.

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

destination-classes

Syntax
```
destination-classes {
    destination-class-name;
}
```

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the destination classes for which statistics are collected.

Options *destination-class-name*—Name of the destination class to include in the source class usage profile.

Usage Guidelines See "Configuring a Class Usage Profile" on page 415.

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

fields

See the following sections:

fields (for Interface Profiles)

Syntax

```
fields {
    field-name;
}
```

Hierarchy Level [edit accounting-options interface-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Statistics to collect in an accounting-data log file for an interface.

Options *field-name*—Name of the field:

- input-bytes—Input bytes
- input-errors—Generic input error packets
- input-multicast—Input packets arriving by multicast
- input-packets—Input packets
- input-unicast—Input unicast packets
- output-bytes—Output bytes
- output-errors—Generic output error packets
- output-multicast—Output packets sent by multicast
- output-packets—Output packets
- output-unicast—Output unicast packets

Usage Guidelines See “Configuring the Interface Profile” on page 405.

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

fields (for Routing Engine Profiles)

Syntax
```
fields {
    field-name;
}
```

Hierarchy Level [edit accounting-options routing-engine-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Statistics to collect in an accounting-data log file for a Routing Engine.

Options *field-name*—Name of the field:

- cpu-load-1—Average system load over the last 1 minute
- cpu-load-5—Average system load over the last 5 minutes
- cpu-load-15—Average system load over the last 15 minutes
- date—Date, in YYYYMMDD format
- host-name—Hostname for the router
- time-of-day—Time of day, in HHMMSS format
- uptime—Time since last reboot, in seconds

Usage Guidelines See “Configuring the Routing Engine Profile” on page 418.

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

file

See the following sections:

file (Associating with a Profile)

Syntax file *filename*;

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*],
[edit accounting-options filter-profile *profile-name*],
[edit accounting-options interface-profile *profile-name*],
[edit accounting-options routing-engine-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description The accounting log file to use.

Options *filename*—Name of the log file. You must specify a *filename* already configured in the file statement at the [edit accounting-options] hierarchy level.

Usage Guidelines See "Configuring the Interface Profile" on page 405, "Configuring the Filter Profile" on page 407, and "Configuring the Routing Engine Profile" on page 418.

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

file (Configuring a Log File)

Syntax
```
file filename {
    archive-sites {
      site-name;
    }
    files filenumber;
    size bytes;
    transfer-interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Information on a log file used for accounting data.

Options *filename*—Name of file in which to write the accounting data.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Files" on page 403.

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

files

Syntax files *number*;

Hierarchy Level [edit accounting-options file *filename*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Information on a log file used for accounting data.

Options *number*—The maximum number of files. When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. The minimum value for *number* is 3 and the default value is 10.

Usage Guidelines See "Configuring Files" on page 403.

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

filter-profile

Syntax
```
filter-profile profile-name {
    counters {
        counter-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a profile to filter and collect packet and byte count statistics and write them to a file in the /var/log directory. To apply the profile to a firewall filter, you include the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level. For more information on firewall filters, see the *JUNOS Network Interfaces Configuration Guide*.

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

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Filter Profile" on page 407.

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

interface-profile

Syntax
```
interface-profile profile-name {
    fields {
      field-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a profile to filter and collect error and packet statistics and write them to a file in the /var/log directory. You can specify an interface profile for either a physical or a logical interface.

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

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Interface Profile" on page 405.

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

interval

Syntax interval *minutes*;

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*],
[edit accounting-options filter-profile *profile-name*],
[edit accounting-options interface-profile *profile-name*],
[edit accounting-options routing-engine-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description How often statistics are collected for the accounting profile.

Options *minutes*—Amount of time between each collection of statistics.
Range: 1 through 2880 minutes
Default: 30 minutes

Usage Guidelines See "Configuring the Interface Profile" on page 405, "Configuring the Filter Profile" on page 407, and "Configuring the Routing Engine Profile" on page 418.

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

routing-engine-profile

Syntax
```
routing-engine-profile profile-name {
    fields {
      field-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a Routing Engine profile to collect selected Routing Engine statistics and write them to a file in the /var/log directory.

Options *profile-name*—Name of the Routing Engine statistics profile.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Routing Engine Profile" on page 418.

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

size

Syntax
```
size bytes;
```

Hierarchy Level [edit accounting-options file *filename*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Attributes of an accounting-data log file.

Options *bytes*—Maximum size of each log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. If you do not specify a size, the file is closed, archived, and renamed when the time specified for the transfer interval is exceeded.
Syntax: *x* to specify bytes, *x*k to specify KB, *x*m to specify MB, *x*g to specify GB
Range: 256 KB through 1 GB

Usage Guidelines See "Configuring the Maximum Size of the File" on page 404.

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

source-classes

Syntax source-classes {
source-class-name;

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the source classes for which statistics are collected.

Options *source-class-name*—Name of the source class to include in the class usage profile.

Usage Guidelines See "Configuring a Class Usage Profile" on page 415.

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

transfer-interval

Syntax transfer-interval *minutes*;

Hierarchy Level [edit accounting-options file *filename*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Time the file remains open and receiving new statistics before it is closed and transferred to an archive site.

Options *minutes*—Time the file remains open and receiving new statistics before it is closed and transferred to an archive site.
Range: 15 through 2880 minutes
Default: 30 minutes

Usage Guidelines See "Configuring the Transfer Interval of the File" on page 404.

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

Part 8
Indexes

Index

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Index of Statements and Commands

O

P

R

S

T

U

V

W