



**JUNOS® Software**

# **Network Management Configuration Guide**

*Release 9.3*

**Juniper Networks, Inc.**

1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA

408-745-2000

**[www.juniper.net](http://www.juniper.net)**

Part Number: 530-027195-01, Revision 1

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, ScreenOS, and Steel-Belted Radius are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOSe is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

*JUNOS® Software Network Management Configuration Guide*

Release 9.3

Copyright © 2008, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Abhilash Prabhakaran, Chris Miller

Editing: Benjamin Mann, Stella Hackell, Nancy Kurahashi, Sonia Saruba, Joanne McClintock

Illustration: Faith Bradford

Cover Design: Edmonds Design

Revision History

10 October 2008—Revision 1

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

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.

## 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 (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) 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, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

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 Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

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; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

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, realms, devices, links, ports 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. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

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 Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) 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 payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

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 présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés 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

	About This Guide	xli
<b>Part 1</b>	<b>Network Management Introduction</b>	
Chapter 1	Network Management Overview	3
Chapter 2	Complete Network Management Configuration Statements	7
<b>Part 2</b>	<b>Integrated Local Management Interface</b>	
Chapter 3	Integrated Local Management Interface Overview	15
<b>Part 3</b>	<b>SNMP</b>	
Chapter 4	SNMP Overview	19
Chapter 5	Configuring SNMP	31
Chapter 6	SNMPv3 Overview	51
Chapter 7	Configuring SNMPv3	53
Chapter 8	SNMP Remote Operations	87
Chapter 9	SNMP Support for Routing Instances	105
Chapter 10	Juniper Networks Enterprise-Specific MIBs	123
Chapter 11	Juniper Networks Enterprise-Specific SNMP Traps	131
Chapter 12	Standard SNMP Traps	143
Chapter 13	Summary of SNMP Configuration Statements	165
Chapter 14	Summary of SNMPv3 Configuration Statements	183
<b>Part 4</b>	<b>RMON Alarms and Events</b>	
Chapter 15	Configuring RMON Alarms and Events	221
Chapter 16	Monitoring RMON Alarms and Events	229
Chapter 17	Summary of RMON Alarm and Event Configuration Statements	239
<b>Part 5</b>	<b>Health Monitoring</b>	
Chapter 18	Configuring Health Monitoring	251
Chapter 19	Summary of Health Monitoring Configuration Statements	257
<b>Part 6</b>	<b>Monitoring Service Quality</b>	
Chapter 20	Monitoring Service Quality in Service Provider Networks	263

**Part 7****Juniper Networks Enterprise-Specific MIBs**

Chapter 21	Interpreting the Structure of Management Information MIB	289
Chapter 22	Interpreting the Enterprise-Specific Chassis MIBs	295
Chapter 23	Interpreting the Enterprise-Specific Destination Class Usage MIB	387
Chapter 24	Interpreting the Enterprise-Specific BGP4 V2 MIB	389
Chapter 25	Interpreting the Enterprise-Specific Ping MIB	391
Chapter 26	Interpreting the Enterprise-Specific Traceroute MIB	405
Chapter 27	Interpreting the Enterprise-Specific RMON Events and Alarms MIB	407
Chapter 28	Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB	411
Chapter 29	Interpreting the Enterprise-Specific Source Class Usage MIB	413
Chapter 30	Interpreting the Enterprise-Specific Passive Monitoring MIB	415
Chapter 31	Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB	417
Chapter 32	Interpreting the Enterprise-Specific SONET APS MIB	419
Chapter 33	Interpreting the Enterprise-Specific IPSec Monitoring MIB	429
Chapter 34	Interpreting the Enterprise-Specific Ethernet MAC MIB	437
Chapter 35	Interpreting the Enterprise-Specific Interface MIB	439
Chapter 36	Interpreting the Enterprise-Specific VPN MIB	445
Chapter 37	Interpreting the Enterprise-Specific Flow Collection Services MIB	457
Chapter 38	Interpreting the Enterprise-Specific Services PIC MIB	461
Chapter 39	Interpreting the Enterprise-Specific Dynamic Flow Capture MIB	467
Chapter 40	Interpreting the Enterprise-Specific Chassis Forwarding MIB	475
Chapter 41	Interpreting the Enterprise-Specific System Log MIB	477
Chapter 42	Interpreting the Enterprise-Specific MPLS LDP MIB	481
Chapter 43	Interpreting the Enterprise-Specific Packet Forwarding Engine MIB	483
Chapter 44	Interpreting the Enterprise-Specific Event MIB	487
Chapter 45	Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB	489
Chapter 46	Interpreting the Enterprise-Specific Layer 2 Transport Protocol (L2TP) MIB	491
Chapter 47	Interpreting the Enterprise-Specific Real-Time Performance Monitoring (RPM) MIB	501
Chapter 48	Interpreting the Enterprise-Specific Class-of-Service MIB	509
Chapter 49	Interpreting the Enterprise-Specific IP Forward MIB	513
Chapter 50	Interpreting the Enterprise-Specific ATM Class-of-Service MIB	515
Chapter 51	Interpreting the Enterprise-Specific Firewall MIB	521
Chapter 52	Interpreting the Enterprise-Specific ATM MIB	523
Chapter 53	Interpreting the Enterprise-Specific Configuration Management MIB	533
Chapter 54	Interpreting the Enterprise-Specific IPv4 MIB	537
Chapter 55	Interpreting the Enterprise-Specific Alarm MIB	539
Chapter 56	Interpreting the Enterprise-Specific Resource Reservation Protocol (RSVP) MIB	541
Chapter 57	Interpreting the Enterprise-Specific MPLS MIB	543
Chapter 58	Interpreting the Enterprise-Specific Host Resources MIB	549



Chapter 59	Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB	551
Chapter 60	Interpreting the Enterprise-Specific MIMSTP MIB	553
Chapter 61	Interpreting the Enterprise-Specific L2ALD MIB	567
Chapter 62	Interpreting the Enterprise-Specific Utility MIB	569
Chapter 63	Interpreting the Enterprise-Specific AAA Objects MIB	573
Chapter 64	Interpreting the Enterprise-Specific Access Authentication Objects MIB	577
Chapter 65	Interpreting the Enterprise-Specific DNS Objects MIB	579
Chapter 66	Interpreting the Enterprise-Specific IPSec Generic Flow Monitoring Object MIB	581
Chapter 67	Interpreting the Enterprise-Specific IPSec VPN Objects MIB	595
Chapter 68	Interpreting the Enterprise-Specific Network Address Translation Objects MIB	599
Chapter 69	Interpreting the Enterprise-Specific Policy Objects MIB	603
Chapter 70	Interpreting the Enterprise-Specific Security Interface Extension Objects MIB	609
Chapter 71	Interpreting the VPN Certificate Objects MIB	613
Chapter 72	Interpreting the Enterprise-Specific Security Screening Objects MIB	615
Chapter 73	Interpreting the Enterprise-Specific LDP MIB	631
Chapter 74	Interpreting the Enterprise-Specific EX-Series SMI MIB	635
Chapter 75	Interpreting the Enterprise-Specific Analyzer MIB	637
Chapter 76	Interpreting the Enterprise-Specific VLAN MIB	641
Chapter 77	Interpreting the Enterprise-Specific Virtual Chassis MIB	645
Chapter 78	Interpreting the Enterprise-Specific PAE Extension MIB	647
Chapter 79	Interpreting the Enterprise-Specific Secure Access Port MIB	651

## Part 8

### Accounting Options

Chapter 80	Accounting Options Overview	657
Chapter 81	Configuring Accounting Options	659
Chapter 82	Summary of Accounting Options Configuration Statements	683

## Part 9

### Index

Index	701
Index of Statements and Commands	711



# Table of Contents

	<b>About This Guide</b>	<b>xli</b>
	Objectives .....	xli
	Audience .....	xli
	Supported Routing Platforms .....	xl ii
	Using the Indexes .....	xl ii
	Using the Examples in This Manual .....	xl ii
	Merging a Full Example .....	xl iii
	Merging a Snippet .....	xl iii
	Documentation Conventions .....	xl iv
	List of Technical Publications .....	xl vi
	Documentation Feedback .....	liii
	Requesting Technical Support .....	liii
<b>Part 1</b>	<b>Network Management Introduction</b>	
<b>Chapter 1</b>	<b>Network Management Overview</b>	<b>3</b>
	Understanding the JUNOS Device Management Functions .....	3
<b>Chapter 2</b>	<b>Complete Network Management Configuration Statements</b>	<b>7</b>
	Configuration Statements at the [edit accounting-options] Hierarchy Level .....	7
	Configuration Statements at the [edit snmp] Hierarchy Level .....	8
<b>Part 2</b>	<b>Integrated Local Management Interface</b>	
<b>Chapter 3</b>	<b>Integrated Local Management Interface Overview</b>	<b>15</b>
	Understanding Integrated Local Management Interface .....	15

**Part 3****SNMP****Chapter 4****SNMP Overview****19**

Understanding SNMP Implementation in the JUNOS Software .....	19
SNMP Architecture .....	19
Management Information Base .....	20
SNMP Traps and Informs .....	20
JUNOS SNMP Agent Features .....	22
Standard SNMP MIBs Supported by the JUNOS Software .....	22

**Chapter 5****Configuring SNMP****31**

Configuring SNMP on a JUNOS Device .....	32
Configuring the System Contact on a JUNOS Device .....	34
Configuring the System Location for a JUNOS Device .....	34
Configuring the System Description on a JUNOS Device .....	34
Filtering Duplicate SNMP Requests .....	35
Configuring the Commit Delay Timer .....	35
Configuring the System Name .....	35
Configuring the SNMP Community String .....	36
Adding a Group of Clients to an SNMP Community .....	37
Configuring SNMP Trap Options and Groups on a JUNOS Device .....	38
Configuring SNMP Trap Options .....	39
Configuring the Source Address for SNMP Traps .....	40
Configuring the Agent Address for SNMP Traps .....	41
Configuring SNMP Trap Groups .....	41
Configuring the Interfaces on Which SNMP Requests Can Be Accepted .....	44
Configuring MIB Views .....	45
Tracing SNMP Activity on a JUNOS Device .....	46
Configuring the SNMP Log Filename .....	47
Configuring the Number and Size of SNMP Log Files .....	47
Configuring Access to the Log File .....	47
Configuring a Regular Expression for Lines to Be Logged .....	47
Configuring the Trace Operations .....	48
Configuring the Local Engine ID .....	49

**Chapter 6****SNMPv3 Overview****51**

SNMPv3 Overview .....	51
-----------------------	----

**Chapter 7****Configuring SNMPv3****53**

Complete SNMPv3 Configuration Statements .....	54
Minimum SNMPv3 Configuration on a JUNOS Device .....	55
Configuring the Local Engine ID .....	56

Creating SNMPv3 Users .....	57
Configuring the SNMPv3 Authentication Type .....	58
Configuring MD5 Authentication .....	58
Configuring SHA Authentication .....	58
Configuring No Authentication .....	59
Configuring the Encryption Type .....	59
Configuring the Advanced Encryption Standard Algorithm .....	59
Configuring the Data Encryption Algorithm .....	60
Configuring Triple DES .....	60
Configuring No Encryption .....	60
Example: Creating SNMPv3 Users Configuration .....	61
Defining Access Privileges for an SNMP Group .....	62
Configuring the Access Privileges Granted to a Group .....	63
Configuring the Group .....	63
Configuring the Security Model .....	63
Configuring the Security Level .....	63
Associating MIB Views with an SNMP User Group .....	64
Configuring the Notify View .....	65
Configuring the Read View .....	65
Configuring the Write View .....	65
Example: Access Privilege Configuration .....	65
Assigning Security Names to Groups .....	66
Configuring the Security Model .....	66
Configuring the Security Name .....	67
Configuring the Group .....	67
Example: Security Group Configuration .....	68
Configuring SNMPv3 Traps on a JUNOS Device .....	68
Configuring the SNMPv3 Trap Notification .....	69
Configuring the Trap Notification Filter .....	70
Configuring the Trap Target Address .....	70
Configuring the Address .....	71
Configuring the Address Mask .....	71
Configuring the Port .....	72
Configuring the Routing Instance .....	72
Configuring the Tag List .....	72
Applying Target Parameters .....	73
Defining and Configuring the Trap Target Parameters .....	74
Applying the Trap Notification Filter .....	74
Configuring the Target Parameters .....	74
Configuring the Message Processing Model .....	75
Configuring the Security Model .....	75
Configuring the Security Level .....	75
Configuring the Security Name .....	76
Configuring SNMP Informs .....	76
Configuring the Remote Engine and Remote User .....	77
Example: Configuring the Remote Engine ID and Remote Users .....	78
Configuring the Inform Notification Type and Target Address .....	78

Example: Configuring the Inform Notification Type and Target Address .....	80
Configuring the SNMPv3 Community .....	80
Configuring the Community Name .....	81
Configuring the Security Names .....	81
Configuring the Tag .....	82
Example: SNMPv3 Community Configuration .....	82
Example: SNMPv3 Configuration .....	82

## Chapter 8

### SNMP Remote Operations

**87**

SNMP Remote Operations Overview .....	87
SNMP Remote Operation Requirements .....	88
Setting SNMP Views .....	88
Example: Setting SNMP Views .....	88
Setting Trap Notification for Remote Operations .....	89
Example: Setting Trap Notification for Remote Operations .....	89
Using Variable-Length String Indexes .....	89
Example: Set Variable-Length String Indexes .....	89
Enabling Logging .....	90
Using the Ping MIB .....	90
Starting a Ping Test .....	90
Using Multiple Set PDUs .....	91
Using a Single Set PDU .....	91
Monitoring a Running Ping Test .....	91
pingResultsTable .....	92
pingProbeHistoryTable .....	93
Generating Traps .....	94
Gathering Ping Test Results .....	94
Stopping a Ping Test .....	96
Interpreting Ping Variables .....	96
Using the Traceroute MIB .....	97
Starting a Traceroute Test .....	97
Using Multiple Set PDUs .....	98
Using a Single Set PDU .....	98
Monitoring a Running Traceroute Test .....	98
traceRouteResultsTable .....	98
traceRouteProbeResultsTable .....	99
traceRouteHopsTable .....	101
Generating Traps .....	102
Monitoring Traceroute Test Completion .....	102
Gathering Traceroute Test Results .....	103
Stopping a Traceroute Test .....	104
Traceroute Variables .....	104

## Chapter 9

### SNMP Support for Routing Instances

**105**

Understanding SNMP Support for Routing Instances .....	105
Support Classes for MIB Objects .....	106

	Identifying a Routing Instance .....	107
	Enabling SNMP Access over Routing Instances .....	108
	Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community .....	108
	Example: Configuring Interface Settings for a Routing Instance .....	109
	Configuring Access Lists for SNMP Access over Routing Instances .....	110
	Trap Support for Routing Instances .....	111
	MIB Support Details .....	111
<b>Chapter 10</b>	<b>Juniper Networks Enterprise-Specific MIBs</b>	<b>123</b>
	Juniper Networks Enterprise-Specific MIBs .....	123
<b>Chapter 11</b>	<b>Juniper Networks Enterprise-Specific SNMP Traps</b>	<b>131</b>
	Juniper Networks Enterprise-Specific SNMP Version 1 Traps .....	131
	Juniper Networks Enterprise-Specific SNMP Version 2 Traps .....	135
	Juniper Networks Enterprise-Specific LDP Traps .....	139
	Disabling LDP Traps .....	139
	Juniper Networks Enterprise-Specific Version 2 Traps on EX-Series Ethernet Switches .....	139
	Juniper Networks Enterprise-Specific Version 2 Traps on MX960 Platforms .....	139
	Raising Traps for Events Based on System Log Messages .....	140
	Unsupported Enterprise-Specific SNMP Traps .....	140
	Spoofing Enterprise-Specific SNMP Traps .....	141
<b>Chapter 12</b>	<b>Standard SNMP Traps</b>	<b>143</b>
	Standard SNMP Version 1 Traps .....	143
	SNMP Version 1 Standard Traps .....	145
	SNMP Version 1 Ping Traps MIB .....	146
	SNMP Version 1 Traceroute Traps MIB .....	147
	SNMP Version 1 VRRP Traps MIB .....	148
	Standard SNMP Version 2 Traps .....	149
	SNMP Version 2 Standard Traps .....	151
	SNMP Version 2 MPLS Traps .....	152
	SNMP Version 2 OSPF Traps MIB .....	153
	SNMP Version 2 Ping Traps MIB .....	157
	SNMP Version 2 Traceroute Traps MIB .....	158
	SNMP Version 2 VRRP Traps MIB .....	159
	Standard SNMP Traps on EX-Series Ethernet Switches .....	159
	Unsupported Standard SNMP Traps .....	160
	Spoofing Standard SNMP Traps .....	164
<b>Chapter 13</b>	<b>Summary of SNMP Configuration Statements</b>	<b>165</b>
	agent-address .....	165
	authorization .....	166

categories .....	166
client-list .....	167
client-list-name .....	167
clients .....	168
commit-delay .....	168
community .....	169
contact .....	170
description .....	170
destination-port .....	171
engine-id .....	171
filter-duplicates .....	171
interface .....	172
location .....	172
logical-system .....	173
name .....	173
nonvolatile .....	174
oid .....	174
routing-instance .....	175
snmp .....	175
source-address .....	176
targets .....	176
traceoptions .....	177
trap-group .....	179
trap-options .....	180
version .....	180
view .....	181
view (Associating MIB View with a Community) .....	181
view (Configuring MIB View) .....	182

**Chapter 14****Summary of SNMPv3 Configuration Statements****183**

address .....	183
address-mask .....	184
authentication-md5 .....	184
authentication-none .....	185
authentication-password .....	185
authentication-sha .....	186
community-name .....	187
engine-id .....	188
group .....	189
group (Configuring) .....	189
group (Defining Access Privileges for an SNMPv3 Group) .....	189
inform-retry-count .....	190
inform-timeout .....	190
local-engine .....	191
message-processing-model .....	192
notify .....	192
notify-filter .....	193
notify-filter (Applying to Management Target) .....	193
notify-filter (Configuring) .....	193



notify-view .....	194
oid .....	194
parameters .....	195
port .....	195
privacy-3des .....	196
privacy-aes128 .....	197
privacy-des .....	198
privacy-none .....	198
privacy-password .....	199
read-view .....	199
remote-engine .....	200
routing-instance .....	201
security-level .....	202
security-level (Defining Access Privileges) .....	202
security-level (Generating SNMP Notifications) .....	202
security-model .....	203
security-model (Access Privileges) .....	203
security-model (Group) .....	203
security-model (SNMP Notifications) .....	204
security-name .....	205
security-name (Community String) .....	205
security-name (Security Group) .....	206
security-name (SNMP Notifications) .....	206
security-to-group .....	207
snmp-community .....	207
tag .....	208
tag-list .....	208
target-address .....	209
target-parameters .....	210
type .....	210
user .....	211
usm .....	212
v3 .....	214
vacm .....	216
view .....	217
write-view .....	218

## Part 4

## RMON Alarms and Events

### Chapter 15

### Configuring RMON Alarms and Events

**221**

Understanding RMON Alarms and Events Configuration .....	221
Minimum RMON Alarm and Event Entry Configuration .....	222
Configuring an Alarm Entry and Its Attributes .....	222
Configuring the Alarm Entry .....	223
Configuring the Description .....	223
Configuring the Falling Event Index or Rising Event Index .....	223
Configuring the Falling Threshold or Rising Threshold .....	224

Configuring the Interval .....	224
Configuring the Falling Threshold Interval .....	224
Configuring the Request Type .....	225
Configuring the Sample Type .....	225
Configuring the Startup Alarm .....	226
Configuring the System Log Tag .....	226
Configuring the Variable .....	226
Configuring an Event Entry and Its Attributes .....	226
Example: Configuring an RMON Alarm and Event Entry .....	227

**Chapter 16****Monitoring RMON Alarms and Events 229**

RMON Alarms .....	229
alarmTable .....	230
jnxRmonAlarmTable .....	230
Using alarmTable to Monitor MIB Objects .....	231
Creating an Alarm Entry .....	231
Configuring the Alarm MIB Objects .....	231
alarmInterval .....	232
alarmVariable .....	232
alarmSampleType .....	232
alarmValue .....	232
alarmStartupAlarm .....	232
alarmRisingThreshold .....	233
alarmFallingThreshold .....	233
alarmOwner .....	233
alarmRisingEventIndex .....	233
alarmFallingEventIndex .....	233
Activating a New Row in alarmTable .....	234
Modifying an Active Row in alarmTable .....	234
Deactivating a Row in alarmTable .....	234
RMON Events .....	234
eventTable .....	234
Using eventTable to Log Alarms .....	235
Creating an Event Entry .....	235
Configuring the MIB Objects .....	235
Activating a New Row in eventTable .....	237
Deactivating a Row in eventTable .....	237

**Chapter 17****Summary of RMON Alarm and Event Configuration Statements 239**

alarm .....	239
community .....	240
description .....	240
event .....	241
falling-event-index .....	241
falling-threshold .....	242
falling-threshold-interval .....	242
interval .....	243
request-type .....	243

rising-event-index .....	244
rising-threshold .....	244
rmon .....	245
sample-type .....	245
startup-alarm .....	246
syslog-subtag .....	246
type .....	247
variable .....	247

## Part 5

## Health Monitoring

---

### Chapter 18

### Configuring Health Monitoring 251

---

Configuring Health Monitoring on JUNOS Devices .....	251
Monitored Objects .....	252
Minimum Health Monitoring Configuration .....	253
Configuring the Falling Threshold or Rising Threshold .....	253
Configuring the Interval .....	254
Log Entries and Traps .....	254
Example: Configuring Health Monitoring .....	254

### Chapter 19

### Summary of Health Monitoring Configuration Statements 257

---

falling-threshold .....	257
health-monitor .....	258
interval .....	258
rising-threshold .....	259

## Part 6

## Monitoring Service Quality

---

### Chapter 20

### Monitoring Service Quality in Service Provider Networks 263

---

Understanding Measurement Points, Key Performance Indicators, and	
Baseline Values .....	263
Measurement Points .....	263
Basic Key Performance Indicators .....	264
Setting Baselines .....	265
Understanding RMON for Monitoring Service Quality .....	265
Setting Thresholds .....	265
RMON Command-Line Interface .....	266
RMON Event Table .....	267
RMON Alarm Table .....	267
Troubleshooting RMON .....	268

Defining and Measuring Network Availability .....	269
Defining Network Availability .....	269
Monitoring the SLA and the Required Bandwidth .....	271
Measuring Availability .....	271
Real-Time Performance Monitoring .....	272
Measuring Health .....	274
Measuring Performance .....	280
Measuring Class of Service .....	282
Inbound Firewall Filter Counters per Class .....	283
Monitoring Output Bytes per Queue .....	284
Dropped Traffic .....	285

## Part 7

### Juniper Networks Enterprise-Specific MIBs

---

<b>Chapter 21</b>	<b>Interpreting the Structure of Management Information MIB</b>	<b>289</b>
	jnxProducts .....	289
	jnxServices .....	289
	jnxMibs .....	290
	jnxTraps .....	292
	jnxExperiment .....	292
<b>Chapter 22</b>	<b>Interpreting the Enterprise-Specific Chassis MIBs</b>	<b>295</b>
	jnxBoxAnatomy .....	295
	Top-Level Objects .....	296
	jnxContainersTable .....	296
	jnxContentsLastChange .....	303
	jnxContentsTable .....	303
	jnxLEDLastChange .....	314
	jnxLEDTable .....	314
	jnxFilledLastChange .....	317
	jnxFilledTable .....	318
	jnxOperatingTable .....	327
	jnxRedundancyTable .....	335
	jnxFruTable .....	340
	jnxBoxKernelMemoryUsedPercent .....	375
	Chassis Traps .....	375
	SNMPv1 Trap Format .....	377
	SNMPv2 Trap Format .....	378
	Chassis Definitions for Router Model MIB .....	380
	MIB Objects for the M120 Router .....	381
	MIB Objects for the MX960 Ethernet Services Router .....	382
	MIB Objects for the MX480 Ethernet Services Router .....	382
	MIB Objects for the MX240 Ethernet Services Router .....	383
	MIB Objects for the EX-Series Ethernet Switches .....	383
	MIB Objects for the SRX 5600 Services Gateway .....	385
	MIB Objects for the SRX 5800 Services Gateway .....	385

<b>Chapter 23</b>	<b>Interpreting the Enterprise-Specific Destination Class Usage MIB</b>	<b>387</b>
	jnxDCUsTable .....	387
	jnxDcuStatsTable .....	388
<b>Chapter 24</b>	<b>Interpreting the Enterprise-Specific BGP4 V2 MIB</b>	<b>389</b>
	jnxBgpM2PrefixCountersTable .....	389
	JnxBgpM2PrefixCountersEntry .....	389
<b>Chapter 25</b>	<b>Interpreting the Enterprise-Specific Ping MIB</b>	<b>391</b>
	jnxPingCtlTable .....	391
	jnxPingCtlEntry .....	392
	jnxPingResultsTable .....	395
	jnxpingResultsEntry .....	395
	jnxPingProbeHistoryTable .....	398
	jnxPingProbeHistoryEntry .....	398
	jnxPingLastTestResultTable .....	400
	jnxPingLastTestResultEntry .....	400
<b>Chapter 26</b>	<b>Interpreting the Enterprise-Specific Traceroute MIB</b>	<b>405</b>
	jnxTraceRouteCtlTable .....	405
	jnxTraceRouteCtlEntry .....	405
<b>Chapter 27</b>	<b>Interpreting the Enterprise-Specific RMON Events and Alarms MIB</b>	<b>407</b>
	jnxRmonAlarmTable .....	407
	RMON Event and Alarm Traps .....	409
<b>Chapter 28</b>	<b>Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB</b>	<b>411</b>
	jnxRpfStatsTable .....	411
	jnxRpfStatsEntry .....	411
<b>Chapter 29</b>	<b>Interpreting the Enterprise-Specific Source Class Usage MIB</b>	<b>413</b>
	jnxScuStatsTable .....	413
	jnxRpfStatsEntry .....	413
<b>Chapter 30</b>	<b>Interpreting the Enterprise-Specific Passive Monitoring MIB</b>	<b>415</b>
	jnxPMonFlowTable .....	415

<b>Chapter 31</b>	<b>Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB</b>	<b>417</b>
	jnxSonetAlarmsTable .....	417
	jnxSonetAlarmEntry .....	417
<b>Chapter 32</b>	<b>Interpreting the Enterprise-Specific SONET APS MIB</b>	<b>419</b>
	apsConfigTable .....	419
	apsConfigEntry .....	419
	apsStatusTable .....	421
	apsStatusEntry .....	421
	apsChanConfigTable .....	424
	apsChanConfigEntry .....	424
	apsChanStatusTable .....	425
	apsChanStatusEntry .....	425
<b>Chapter 33</b>	<b>Interpreting the Enterprise-Specific IPSec Monitoring MIB</b>	<b>429</b>
	jnxIkeTunnelTable .....	429
	jnxIkeTunnelEntry .....	429
	jnxIPSecTunnelTable .....	432
	jnxIPSecTunnelEntry .....	432
	jnxIPSecSaTable .....	434
	jnxIPSecSaEntry .....	434
<b>Chapter 34</b>	<b>Interpreting the Enterprise-Specific Ethernet MAC MIB</b>	<b>437</b>
	jnxMacStatsTable .....	437
	jnxMacStatsEntry .....	437
<b>Chapter 35</b>	<b>Interpreting the Enterprise-Specific Interface MIB</b>	<b>439</b>
	jnxIfTable .....	439
	jnxIfEntry .....	439
	ifChassisTable .....	441
	ifChassisEntry .....	441
<b>Chapter 36</b>	<b>Interpreting the Enterprise-Specific VPN MIB</b>	<b>445</b>
	jnxVpnInfo .....	445
	jnxVpnTable .....	446
	jnxVpnEntry .....	446
	jnxVpnIfTable .....	447
	jnxVpnIfEntry .....	447
	jnxVpnPwTable .....	450
	jnxVpnPwEntry .....	450

	jnxVpnRTTable .....	455
	jnxVpnRTEntry .....	455
	VPN Traps .....	455
<b>Chapter 37</b>	<b>Interpreting the Enterprise-Specific Flow Collection Services MIB</b>	<b>457</b>
	jnxCollGlobalStats .....	457
	jnxCollPicIfTable .....	458
	jnxCollPicEntry .....	458
	jnxCollFileTable .....	459
	jnxCollFileEntry .....	460
<b>Chapter 38</b>	<b>Interpreting the Enterprise-Specific Services PIC MIB</b>	<b>461</b>
	jnxSpSvcSetTable .....	461
	jnxSpSvcSetEntry .....	461
	jnxSpSvcSetSvcTypeTable .....	463
	jnxSpSvcSetSvcTypeEntry .....	463
	jnxSpSvcSetIfTable .....	463
	jnxSpSvcSetSvcIfEntry .....	464
	Service Traps .....	464
	Redundant Interfaces .....	465
<b>Chapter 39</b>	<b>Interpreting the Enterprise-Specific Dynamic Flow Capture MIB</b>	<b>467</b>
	jnxDfcCSTable .....	467
	jnxDfcCSEntry .....	467
	jnxDfcCDTable .....	471
	jnxDfcCDEntry .....	471
	DFC Notification Variables .....	471
	DFC Notification Definitions .....	472
<b>Chapter 40</b>	<b>Interpreting the Enterprise-Specific Chassis Forwarding MIB</b>	<b>475</b>
	jnxFwddProcess .....	475
<b>Chapter 41</b>	<b>Interpreting the Enterprise-Specific System Log MIB</b>	<b>477</b>
	jnxSyslogTable .....	477
	jnxSyslogEntry .....	477
	jnxSyslogAvTable .....	479
	jnxSyslogEntry .....	480

<b>Chapter 42</b>	<b>Interpreting the Enterprise-Specific MPLS LDP MIB</b>	<b>481</b>
<b>Chapter 43</b>	<b>Interpreting the Enterprise-Specific Packet Forwarding Engine MIB</b>	<b>483</b>
	jnxPfeNotifyGlTable .....	483
	jnxPfeNotifyGlEntry .....	483
	jnxPfeNotifyTypeTable .....	485
	jnxPfeNotifyTypeEntry .....	485
<b>Chapter 44</b>	<b>Interpreting the Enterprise-Specific Event MIB</b>	<b>487</b>
	jnxEventAvTable .....	487
	jnxEventAvEntry .....	487
	Notifications for the Event MIB .....	488
<b>Chapter 45</b>	<b>Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB</b>	<b>489</b>
	jnxBfdSessTable .....	489
	jnxBfdSessEntry .....	489
	Notifications for the BFD MIB .....	490
<b>Chapter 46</b>	<b>Interpreting the Enterprise-Specific Layer 2 Transport Protocol (L2TP) MIB</b>	<b>491</b>
	The L2TP Scalar Status and Statistics Group .....	491
	jnxL2tpTunnelGroupStatsTable .....	492
	jnxL2tpTunnelStatsTable .....	493
	jnxL2tpSessionStatsTable .....	495
	jnxL2tpMlpppBundleStatsTable .....	499
<b>Chapter 47</b>	<b>Interpreting the Enterprise-Specific Real-Time Performance Monitoring (RPM) MIB</b>	<b>501</b>
	jnxRpmResultsSampleTable .....	501
	JnxRpmMeasurementType .....	502
	JnxRpmTimestampType .....	503
	jnxRpmResultsSummaryTable .....	503
	jnxRpmResultsCalculatedTable .....	504
	jnxRpmHistorySampleTable .....	505
	jnxRpmHistorySummaryTable .....	506
	jnxRpmHistoryCalculatedTable .....	506
<b>Chapter 48</b>	<b>Interpreting the Enterprise-Specific Class-of-Service MIB</b>	<b>509</b>
	jnxCosInvQstatTable .....	509



<b>Chapter 49</b>	<b>Interpreting the Enterprise-Specific IP Forward MIB</b>	<b>513</b>
	jnxIpCidrRouteTable .....	513
	jnxIpCidrRouteEntry .....	513
<b>Chapter 50</b>	<b>Interpreting the Enterprise-Specific ATM Class-of-Service MIB</b>	<b>515</b>
	jnxCosAtmVcTable .....	515
	jnxCosAtmVcScTable .....	516
	jnxCosAtmVcQstatsTable .....	518
	jnxCosAtmTrunkTable .....	518
<b>Chapter 51</b>	<b>Interpreting the Enterprise-Specific Firewall MIB</b>	<b>521</b>
	jnxFirewallsTable .....	521
	jnxFirewallCounterTable .....	522
<b>Chapter 52</b>	<b>Interpreting the Enterprise-Specific ATM MIB</b>	<b>523</b>
	jnxAtmIfTable .....	523
	jnxAtmVCTable .....	525
	jnxAtmVpTable .....	528
	jnxAtmTrunkTable .....	530
<b>Chapter 53</b>	<b>Interpreting the Enterprise-Specific Configuration Management MIB</b>	<b>533</b>
	Text Conventions .....	533
	Configuration Change Management Objects and	
	jnxCmCfgChgEventTable .....	534
	jnxCmCfgChgEventTable .....	534
	Rescue Configuration Change Management Objects .....	535
	Configuration Management Notifications .....	536
<b>Chapter 54</b>	<b>Interpreting the Enterprise-Specific IPv4 MIB</b>	<b>537</b>
	jnxIpv4AddrTable .....	537
<b>Chapter 55</b>	<b>Interpreting the Enterprise-Specific Alarm MIB</b>	<b>539</b>
	jnxAlarmRelayMode .....	539
	jnxYellowAlarms .....	539
	jnxRedAlarms .....	540

<b>Chapter 56</b>	<b>Interpreting the Enterprise-Specific Resource Reservation Protocol (RSVP) MIB</b>	<b>541</b>
	jnxRsvpSessionTable .....	541
<b>Chapter 57</b>	<b>Interpreting the Enterprise-Specific MPLS MIB</b>	<b>543</b>
	MPLS Info Table .....	543
	MPLS Traffic Engineering (TE) Info Table .....	544
	mplsAdminGroup .....	544
	mplsLspInfoList .....	544
	Enterprise-Specific MPLS Traps .....	547
<b>Chapter 58</b>	<b>Interpreting the Enterprise-Specific Host Resources MIB</b>	<b>549</b>
	jnxHrStorageTable .....	549
<b>Chapter 59</b>	<b>Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB</b>	<b>551</b>
<b>Chapter 60</b>	<b>Interpreting the Enterprise-Specific MIMSTP MIB</b>	<b>553</b>
	jnxMIDot1sJuniperMstTable .....	553
	Juniper Networks MSTI Bridge Table .....	558
	jnxMIMstVlanInstanceMappingTable .....	560
	jnxMIMstCistPortTable .....	561
	jnxMIMstMstiPortTable .....	564
	Juniper Networks Enterprise-Specific MIMSTP Traps .....	566
<b>Chapter 61</b>	<b>Interpreting the Enterprise-Specific L2ALD MIB</b>	<b>567</b>
	jnxl2aldInterfaceTable .....	567
	MAC Address Limit Traps .....	568
<b>Chapter 62</b>	<b>Interpreting the Enterprise-Specific Utility MIB</b>	<b>569</b>
	jnxUtilCounter32Table .....	569
	jnxUtilCounter64Table .....	570
	jnxUtilIntegerTable .....	570
	jnxUtilUintTable .....	570
	jnxUtilStringTable .....	571
<b>Chapter 63</b>	<b>Interpreting the Enterprise-Specific AAA Objects MIB</b>	<b>573</b>
	Text Conventions .....	573
	jnxUserAAASatTable .....	574

	jnxUserAAAServerName .....	574
	Access Authentication-Related Traps .....	574
<b>Chapter 64</b>	<b>Interpreting the Enterprise-Specific Access Authentication Objects MIB</b>	<b>577</b>
	jnxJsFwAuthStats .....	577
	jnxJsAuthTrapVars .....	578
	jnxJsAuthNotifications .....	578
<b>Chapter 65</b>	<b>Interpreting the Enterprise-Specific DNS Objects MIB</b>	<b>579</b>
	jnxJsDnsProxyDataObjects .....	579
<b>Chapter 66</b>	<b>Interpreting the Enterprise-Specific IPSec Generic Flow Monitoring Object MIB</b>	<b>581</b>
	Branch Tree Objects .....	581
	Text Conventions .....	582
	Number of IKE Tunnels Currently Active .....	585
	IPSec Phase 1 IKE Tunnel Table .....	586
	IPSec Phase 2 IKE Tunnel Table .....	589
	IPSec Phase 2 Security Association Table .....	592
<b>Chapter 67</b>	<b>Interpreting the Enterprise-Specific IPSec VPN Objects MIB</b>	<b>595</b>
	Text Conventions .....	595
	jnxJsIpSecTunnelTable .....	596
<b>Chapter 68</b>	<b>Interpreting the Enterprise-Specific Network Address Translation Objects MIB</b>	<b>599</b>
	Source NAT Table .....	599
	jnxJsNatIfSrcPoolPortTable .....	601
	NAT Trap Definitions .....	601
<b>Chapter 69</b>	<b>Interpreting the Enterprise-Specific Policy Objects MIB</b>	<b>603</b>
	Security Policy Table .....	603
	jnxJsPolicyStatsTable .....	605
<b>Chapter 70</b>	<b>Interpreting the Enterprise-Specific Security Interface Extension Objects MIB</b>	<b>609</b>
	jnxJsIfMonTable .....	609

<b>Chapter 71</b>	<b>Interpreting the VPN Certificate Objects MIB</b>	<b>613</b>
	jnxJsLoadedCaCertTable .....	613
	jnxJsLoadedLocalCertTable .....	614
<b>Chapter 72</b>	<b>Interpreting the Enterprise-Specific Security Screening Objects MIB</b>	<b>615</b>
	jnxJsScreenMonTable .....	615
<b>Chapter 73</b>	<b>Interpreting the Enterprise-Specific LDP MIB</b>	<b>631</b>
	LDP Notification Objects and Notification Types .....	631
	LDP Statistics Table .....	634
<b>Chapter 74</b>	<b>Interpreting the Enterprise-Specific EX-Series SMI MIB</b>	<b>635</b>
<b>Chapter 75</b>	<b>Interpreting the Enterprise-Specific Analyzer MIB</b>	<b>637</b>
	Analyzer Table .....	637
	Analyzer Input Table .....	638
	Analyzer Output Table .....	639
<b>Chapter 76</b>	<b>Interpreting the Enterprise-Specific VLAN MIB</b>	<b>641</b>
	VLAN Configuration Table .....	641
	VLAN Interfaces Table .....	642
	Port Group Table .....	643
	MAC List Table .....	643
<b>Chapter 77</b>	<b>Interpreting the Enterprise-Specific Virtual Chassis MIB</b>	<b>645</b>
	Virtual Chassis Member Table .....	645
<b>Chapter 78</b>	<b>Interpreting the Enterprise-Specific PAE Extension MIB</b>	<b>647</b>
	jnxAuthProfileName .....	647
	Authentication Configuration Extension Table .....	647
	Static MAC List Authentication Bypass Table .....	648
	jnxStaticMacAuthBypassIfTable .....	648
<b>Chapter 79</b>	<b>Interpreting the Enterprise-Specific Secure Access Port MIB</b>	<b>651</b>
	Port Security Table for VLAN .....	651
	Port Security Table for Interface .....	652

Storm Control Table .....	653
DHCP Snooping Notification .....	654
MAC Limit Exceeded Notification .....	654
Storm Event Notification .....	654

## Part 8

## Accounting Options

### Chapter 80

### Accounting Options Overview **657**

Accounting Options Overview .....	657
-----------------------------------	-----

### Chapter 81

### Configuring Accounting Options **659**

Accounting Options Configuration .....	659
Accounting Options—Full Configuration .....	659
Minimum Accounting Options Configuration .....	660
Configuring Files .....	662
Configuring the Storage Location of the File .....	663
Configuring the Maximum Size of the File .....	663
Configuring the Maximum Number of Files .....	663
Configuring the Start Time for File Transfer .....	664
Configuring the Transfer Interval of the File .....	664
Configuring Archive Sites .....	664
Configuring the Interface Profile .....	665
Configuring Fields .....	665
Configuring the File Information .....	665
Configuring the Interval .....	666
Example: Configuring the Interface Profile .....	666
Configuring the Filter Profile .....	667
Configuring the Counters .....	668
Configuring the File Information .....	668
Configuring the Interval .....	668
Example: Configuring a Filter Profile .....	669
Example: Configuring Interface-Specific Firewall Counters and Filter Profiles .....	670
Source Class Usage Options Overview .....	671
Configuring SCU or DCU .....	672
Creating Prefix Route Filters in a Policy Statement .....	672
Applying the Policy to the Forwarding Table .....	672
Enabling Accounting on Inbound and Outbound Interfaces .....	672
Configuring SCU on a Virtual Loopback Tunnel Interface .....	673
Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC .....	674
Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface .....	674
Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface .....	674

Configuring Class Usage Profiles .....	675
Configuring a Class Usage Profile .....	675
Configuring the File Information .....	676
Configuring the Interval .....	676
Creating a Class Usage Profile to Collect Source Class Usage Statistics .....	676
Creating a Class Usage Profile to Collect Destination Class Usage Statistics .....	677
Configuring the MIB Profile .....	677
Configuring the File Information .....	678
Configuring the Interval .....	678
Configuring the MIB Operation .....	678
Configuring MIB Object Names .....	678
Example: Configuring a MIB Profile .....	679
Configuring the Routing Engine Profile .....	679
Configuring Fields .....	680
Configuring the File Information .....	680
Configuring Fields .....	680
Configuring the File Information .....	680
Configuring the Interval .....	680
Example: Configuring a Routing Engine Profile .....	681

## Chapter 82

## Summary of Accounting Options Configuration Statements **683**

accounting-options .....	683
archive-sites .....	684
class-usage-profile .....	685
counters .....	686
destination-classes .....	686
fields .....	687
fields (for Interface Profiles) .....	687
fields (for Routing Engine Profiles) .....	688
file .....	689
file (Associating with a Profile) .....	689
file (Configuring a Log File) .....	690
files .....	690
filter-profile .....	691
interface-profile .....	692
interval .....	693
mib-profile .....	694
nonpersistent .....	694
objects-names .....	695
operation .....	695
routing-engine-profile .....	696
size .....	696
source-classes .....	697
start-time .....	697
transfer-interval .....	698

**Part 9****Index**

---

Index .....	701
Index of Statements and Commands .....	711





# List of Figures

Figure 1: Inform Request and Response .....	77
Figure 2: SNMP Data for Routing Instances .....	106
Figure 3: Network Entry Points .....	264
Figure 4: Setting Thresholds .....	266
Figure 5: Regional Points of Presence .....	269
Figure 6: Measurements to Each Router .....	270
Figure 7: Network Behavior During Congestion .....	283



# List of Tables

Table 1: Notice Icons .....	xliv
Table 2: Text and Syntax Conventions .....	xliv
Table 3: Technical Documentation for Supported Routing Platforms .....	xlvi
Table 4: JUNOS Software Network Operations Guides .....	l
Table 5: JUNOS Software with Enhanced Services Documentation .....	li
Table 6: Additional Books Available Through http://www.juniper.net/books .....	lii
Table 7: JUNOS Device Management Features .....	4
Table 8: Standard MIBs Supported on JUNOS Platforms .....	23
Table 9: SNMP Tracing Flags .....	48
Table 10: Results in pingProbeHistoryTable: After the First Ping Test .....	95
Table 11: Results in pingProbeHistoryTable: After the First Probe of the Second Test .....	95
Table 12: Results in pingProbeHistoryTable: After the Second Ping Test .....	96
Table 13: traceRouteProbeHistoryTable .....	103
Table 14: MIB Support for Routing Instances (Juniper Networks MIBs) .....	111
Table 15: Class 1 MIB Objects (Standard and Juniper MIBs) .....	115
Table 16: Class 2 MIB Objects (Standard and Juniper MIBs) .....	119
Table 17: Class 3 MIB Objects (Standard and Juniper MIBs) .....	120
Table 18: Class 4 MIB Objects (Standard and Juniper MIBs) .....	121
Table 19: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps .....	132
Table 20: Enterprise-Specific Supported SNMP Version 2 Traps .....	135
Table 21: Unsupported Enterprise-Specific SNMP Traps .....	140
Table 22: Standard Supported SNMP Version 1 Traps .....	143
Table 23: Standard Supported SNMP Version 2 Traps .....	149
Table 24: Unsupported Standard SNMP Traps .....	161
Table 25: Monitored Object Instances .....	253
Table 26: RMON Event Table .....	267
Table 27: RMON Alarm Table .....	268
Table 28: jnxRmon Alarm Extensions .....	268
Table 29: Real-Time Performance Monitoring Configuration Options .....	272
Table 30: Health Metrics .....	274
Table 31: Counter Values for vlan-ccc Encapsulation .....	279
Table 32: Performance Metrics .....	280
Table 33: Inbound Traffic Per Class .....	284
Table 34: Inbound Counters .....	284
Table 35: Outbound Counters for ATM Interfaces .....	285
Table 36: Outbound Counters for Non-ATM Interfaces .....	285
Table 37: Dropped Traffic Counters .....	285
Table 38: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router .....	298

Table 39: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router .....	299
Table 40: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router .....	299
Table 41: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router .....	300
Table 42: jnxContainersEntry Objects in the jnxContainersTable of an M5 Router .....	301
Table 43: jnxContainersEntry Objects in the jnxContainersTable of a T640 Routing Node .....	301
Table 44: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router .....	302
Table 45: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router .....	302
Table 46: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router .....	304
Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node .....	307
Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router .....	311
Table 49: jnxLEDEntry Objects in the jnxLEDTable of an M20 Router .....	316
Table 50: jnxLEDEntry Objects in the jnxLEDTable of a T640 Routing Node .....	316
Table 51: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router .....	317
Table 52: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router .....	318
Table 53: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node .....	321
Table 54: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router .....	324
Table 55: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router .....	330
Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node .....	330
Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router .....	333
Table 58: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router .....	337
Table 59: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Routing Node .....	338
Table 60: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router .....	339
Table 61: jnxFruContents Objects in the jnxFruTable of an M10 Router .....	343
Table 62: JnxFruContents Objects in the jnxFruTable of an M20 Router .....	346
Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router .....	349
Table 64: jnxFruContents Objects in the jnxFruTable of an M40 Router .....	356
Table 65: JnxFruContents Objects in the jnxFruTable of an M40e Router .....	361
Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node .....	366
Table 67: SNMP Version 1 Trap Format .....	377
Table 68: SNMP Version 2 Trap Format .....	378
Table 69: Router Models and Their sysObjectIds .....	380
Table 70: jnxDCUsEntry .....	387

Table 71: jnxDCUsStatusEntry .....	388
Table 72: jnxBgpM2PrefixCountersEntry .....	389
Table 73: jnxPingCtlEntry .....	392
Table 74: jnxPingsResultsEntry .....	396
Table 75: jnxPingProbeHistoryEntry .....	399
Table 76: jnxPingLastTestResultEntry .....	401
Table 77: jnxTraceRouteCtlTable .....	405
Table 78: jnxRmonAlarmEntry .....	407
Table 79: RMON Event and Alarm Traps .....	409
Table 80: jnxRpfStatsEntry .....	411
Table 81: jnxRpfStatsEntry .....	414
Table 82: jnxPMFlowEntry .....	416
Table 83: jnxSonetAlarmTable .....	417
Table 84: jnxSonetAlarmInterface Objects in the jnxSonetAlarmTable of an M20 Router .....	418
Table 85: apsConfigTable .....	420
Table 86: apsStatusTable .....	421
Table 87: apsChanConfigTable .....	424
Table 88: apsChanStatusTable .....	426
Table 89: jnxIkeTunnelTable .....	430
Table 90: jnxIPSecTunnelTable .....	432
Table 91: jnxIPSecSaTable .....	434
Table 92: jnxMacStatsTable .....	437
Table 93: jnxIfTable .....	439
Table 94: ifChassisTable .....	442
Table 95: Supported jnxVpnInfo Objects, VPNs, and Circuit Connection Services .....	446
Table 96: Supported jnxVpnEntry Objects, VPNs, and Circuit Connection Services .....	446
Table 97: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services .....	448
Table 98: Supported jnxVpnEntry Objects, VPNs, and Connection Circuit Services .....	451
Table 99: Supported jnxVpnRTEntry Objects, VPNs, and Circuit Connection Services .....	455
Table 100: Supported VPN Traps, VPNs, and Circuit Connection Services .....	456
Table 101: jnxCollGlobalStats .....	457
Table 102: jnxCollPicEntry .....	458
Table 103: jnxCollFileTable .....	460
Table 104: jnxSpSvcSetTable .....	462
Table 105: jnxSpSvcSetSvcTypeTable .....	463
Table 106: jnxSpSvcSetIfTable .....	464
Table 107: Supported Traps for Services PIC MIB .....	465
Table 108: jnxDfcCSTable .....	468
Table 109: jnxDfcCDTable .....	471
Table 110: Supported Notification Variables for the DFC MIB .....	471
Table 111: Supported Notification Definitions for the DFC MIB .....	472
Table 112: jnxFwddProcess .....	475
Table 113: jnxSyslogTable .....	478
Table 114: Facilities That Generate System Log Messages .....	478
Table 115: jnxSyslogAvTable .....	480

Table 116: jnxPfeNotifyGlEntry .....	483
Table 117: PFE Notification Types .....	485
Table 118: jnxPfeNotifyTypeTable .....	485
Table 119: jnxEventAvTable .....	487
Table 120: Supported Notifications for the Event MIB .....	488
Table 121: jnxBfdSessTable .....	489
Table 122: Supported Notifications for the BFD MIB .....	490
Table 123: The L2TP Scalar Status and Statistics Group .....	491
Table 124: jnxL2tpTunnelGroupStatsTable .....	492
Table 125: jnxL2tpTunnelStatsTable .....	493
Table 126: jnxL2tpSessionStatsTable .....	495
Table 127: jnxL2tpMlpppBundleStatsTable .....	499
Table 128: jnxRpmResultsSampleTable .....	502
Table 129: JnxRpmMeasurementType .....	502
Table 130: jnxRpmResultsSummaryTable .....	504
Table 131: jnxRpmResultsCalculatedTable .....	505
Table 132: jnxRpmHistorySampleTable .....	505
Table 133: jnxRpmHistorySummaryTable .....	506
Table 134: jnxRpmHistoryCalculatedTable .....	506
Table 135: jnxCosInvQstatEntry .....	509
Table 136: jnxIpCidrRouteTable .....	514
Table 137: jnxCosAtmVcScTable .....	516
Table 138: jnxCosAtmVcQstatsTable .....	518
Table 139: jnxCosAtmTrunkTable .....	519
Table 140: jnxFirewallsEntry .....	521
Table 141: JnxFirewallCounterEntry .....	522
Table 142: jnxAtmIfTable .....	524
Table 143: jnxAtmVCTable .....	526
Table 144: jnxAtmVpTable .....	529
Table 145: jnxAtmTrunkTable .....	530
Table 146: Text Conventions for Enterprise-Specific Configuration Management MIB .....	533
Table 147: Configuration Change Management Objects .....	534
Table 148: jnxCmCfgChgEventTable .....	535
Table 149: Rescue Configuration Change Management Objects .....	535
Table 150: jnxIpv4AddrTable .....	537
Table 151: jnxYellowAlarms .....	540
Table 152: jnxRedAlarms .....	540
Table 153: jnxRsvpSessionTable .....	542
Table 154: mplsInfo .....	543
Table 155: mplsTEInfo .....	544
Table 156: MplsLspInfoEntry .....	545
Table 157: MPLS Traps .....	547
Table 158: jnxMIDot1sJuniperMstTable .....	554
Table 159: jnxMIMstMstiBridgeTable .....	558
Table 160: jnxMIMstVlanInstanceMappingTable .....	560
Table 161: jnxMIMstCistPortTable .....	561
Table 162: jnxMIMstMstiPortTable .....	564
Table 163: Juniper Networks Enterprise-Specific MIMSTP Traps .....	566
Table 164: jnxI2aldInterfaceTable .....	567
Table 165: jnxUtilCounter32Entry .....	570

Table 166: jnxUtilCounter64Entry .....	570
Table 167: jnxUtilIntegerEntry .....	570
Table 168: jnxUtilUintEntry .....	571
Table 169: jnxUtilStringEntry .....	571
Table 170: JnxAuthenticateType .....	573
Table 171: jnxUserAAASatTable .....	574
Table 172: Access Authentication-Related Traps .....	574
Table 173: jnxJsFwAuthStats .....	577
Table 174: jnxJsAuthTrapVars .....	578
Table 175: jnxJsAuthNotifications .....	578
Table 176: jnxJsDnsProxyDataObjects .....	579
Table 177: IKE Identity Type Text Conventions .....	582
Table 178: IKE Negotiation Mode Text Conventions .....	582
Table 179: IKE Negotiations Hash Algorithms .....	582
Table 180: IKE Authentication Method .....	583
Table 181: Role of Local Endpoint in Negotiations .....	583
Table 182: State of Phase 1 IKE Negotiation .....	583
Table 183: Diffie-Hellman Group in Negotiations .....	584
Table 184: Key Used by IPSec Phase 2 Tunnel .....	584
Table 185: Encryption Algorithm in Negotiations .....	584
Table 186: Role of Local Endpoint in Negotiations .....	585
Table 187: Type of Remote Peer Gateway .....	585
Table 188: Role of Local Endpoint in Negotiations .....	585
Table 189: Number of IKE Tunnels Currently Active .....	586
Table 190: IPSec Phase 1 IKE Tunnel Table .....	586
Table 191: IPSec Phase 2 IKE Tunnel Table .....	590
Table 192: IPSec Phase 2 Security Association Table .....	593
Table 193: JnxJsIpSecVpnType .....	595
Table 194: jnxJsIpSecTunnelTable .....	596
Table 195: Source NAT Table .....	599
Table 196: jnxJsNatIfSrcPoolPortTable .....	601
Table 197: NAT Trap Definitions .....	602
Table 198: Security Policy Table .....	603
Table 199: jnxJsPolicyStatsTable .....	606
Table 200: jnxJsIfMonTable .....	610
Table 201: jnxJsLoadedCaCertTable .....	613
Table 202: jnxJsLoadedLocalCertTable .....	614
Table 203: jnxJsScreenMonTable .....	617
Table 204: LDP Notification Objects .....	632
Table 205: LDP Notification Types .....	633
Table 206: jnxLdpStatsTable .....	634
Table 207: jnxExSwitching .....	635
Table 208: jnxAnalyzerTable .....	637
Table 209: jnxAnalyzerInputTable .....	638
Table 210: jnxAnalyzerOutputTable .....	639
Table 211: jnxVlanTable .....	642
Table 212: jnxVlanInterfaceTable .....	642
Table 213: jnxVlanPortGroupTable .....	643
Table 214: jnxVlanMacListTable .....	644
Table 215: jnxVirtualChassisMemberTable .....	645
Table 216: jnxPaeAuthConfigTable .....	647

Table 217: jnxStaticMacAuthBypassTable .....	648
Table 218: jnxSecAccessPortVlanTable .....	652
Table 219: jnxSecAccessPortIfTable .....	652
Table 220: jnxStormCtlTable .....	653
Table 221: Types of Accounting Profiles .....	657



# About This Guide

This preface provides the following guidelines for using the *JUNOS® Software Network Management Configuration Guide*:

- Objectives on page xli
- Audience on page xli
- Supported Routing Platforms on page xlii
- Using the Indexes on page xlii
- Using the Examples in This Manual on page xlii
- Documentation Conventions on page xliv
- List of Technical Publications on page xlv
- Documentation Feedback on page liii
- Requesting Technical Support on page liii

## Objectives

---

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



**NOTE:** This guide documents Release 9.3 of the JUNOS 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 M-series, MX-series, T-series, EX-series, or J-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.

## Supported Routing Platforms

---

For the features described in this manual, the JUNOS software currently supports the following routing platforms:

- J-series
- M-series
- MX-series
- T-series
- EX-series

## Using the Indexes

---

This reference 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.

## Using the Examples in This Manual

---

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

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

## Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

## Merging a Snippet

To merge a snippet, follow these steps:

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

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

```
commit {
```

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

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

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

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

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

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

## Documentation Conventions

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

**Table 1: Notice Icons**





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

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

**Table 2: Text and Syntax Conventions**

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command:  user@host> <b>configure</b>

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

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

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

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select <b>Protocols &gt; Ospf</b> .

## List of Technical Publications

Table 3 on page xlvii lists the software and hardware guides and release notes for Juniper Networks J-series, M-series, MX-series, and T-series routing platforms and describes the contents of each document. Table 4 on page i lists the books included in the *Network Operations Guide* series. Table 5 on page li lists the manuals and release notes supporting JUNOS software with enhanced services. All documents are available at <http://www.juniper.net/techpubs/>.

Table 6 on page lii lists additional books on Juniper Networks solutions that you can order through your bookstore. A complete list of such books is available at <http://www.juniper.net/books>.

**Table 3: Technical Documentation for Supported Routing Platforms**

Book	Description
<b>JUNOS Software for Supported Routing Platforms</b>	
<i>Access Privilege</i>	Explains how to configure access privileges in user classes by using permission flags and regular expressions. Lists the permission flags along with their associated command-line interface (CLI) operational mode commands and configuration statements.
<i>Class of Service</i>	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.
<i>CLI User Guide</i>	Describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms. This material was formerly covered in the <i>JUNOS System Basics Configuration Guide</i> .
<i>Feature Guide</i>	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
<i>High Availability</i>	Provides an overview of hardware and software resources that ensure a high level of continuous routing platform operation and describes how to configure high availability (HA) features such as nonstop active routing (NSR) and graceful Routing Engine switchover (GRES).
<i>MPLS Applications</i>	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.

**Table 3: Technical Documentation for Supported Routing Platforms** (*continued*)

Book	Description
<i>Multicast Protocols</i>	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
<i>Multiplay Solutions</i>	Describes how you can deploy IPTV and voice over IP (VoIP) services in your network.
<i>MX-series Layer 2 Configuration Guide</i>	Provides an overview of the Layer 2 functions of the MX-series routers, including configuring bridging domains, MAC address and VLAN learning and forwarding, and spanning-tree protocols. It also details the routing instance types used by Layer 2 applications. All of this material was formerly covered in the <i>JUNOS Routing Protocols Configuration Guide</i> .
<i>MX-series Layer 2 Solutions Guide</i>	Describes common configuration scenarios for the Layer 2 features supported on the MX-series routers, including basic bridged VLANs with normalized VLAN tags, aggregated Ethernet links, bridge domains, Multiple Spanning Tree Protocol (MSTP), and integrated routing and bridging (IRB).
<i>Network Interfaces</i>	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
<i>Network Management</i>	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
<i>Policy Framework</i>	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, and forwarding options.
<i>Protected System Domain</i>	Provides an overview of the JCS 1200 platform and the concept of Protected System Domains (PSDs). The JCS 1200 platform, which contains up to six redundant pairs of Routing Engines running JUNOS software, is connected to a T320 router or to a T640 or T1600 routing node. To configure a PSD, you assign any number of Flexible PIC concentrators (FPCs) in the T-series routing platform to a pair of Routing Engines on the JCS 1200 platform. Each PSD has the same capabilities and functionality as a physical router, with its own control plane, forwarding plane, and administration.
<i>Routing Protocols</i>	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
<i>Secure Configuration Guide for Common Criteria and JUNOS-FIPS</i>	Provides an overview of secure Common Criteria and JUNOS-FIPS protocols for the JUNOS software and describes how to install and configure secure Common Criteria and JUNOS-FIPS on a routing platform.
<i>Services Interfaces</i>	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the router.
<i>Software Installation and Upgrade Guide</i>	Describes the JUNOS software components and packaging and explains how to initially configure, reinstall, and upgrade the JUNOS system software. This material was formerly covered in the <i>JUNOS System Basics Configuration Guide</i> .

**Table 3: Technical Documentation for Supported Routing Platforms** (*continued*)

Book	Description
<i>Subscriber Access</i>	Provides an overview of the subscriber access features of the JUNOS software and describes how to configure subscriber access support on the router, including dynamic profiles, class of service, AAA, and access methods.
<i>System Basics</i>	Describes Juniper Networks routing platforms and explains how to configure basic system parameters, supported protocols and software processes, authentication, and a variety of utilities for managing your router on the network.
<i>VPNs</i>	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.
<b>JUNOS References</b>	
<i>Hierarchy and RFC Reference</i>	Describes the JUNOS configuration mode commands. Provides a hierarchy reference that displays each level of a configuration hierarchy, and includes all possible configuration statements that can be used at that level. This material was formerly covered in the <i>JUNOS System Basics Configuration Guide</i> .
<i>Interfaces Command Reference</i>	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
<i>Routing Protocols and Policies Command Reference</i>	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing policies and protocols, including firewall filters.
<i>System Basics and Services Command Reference</i>	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 class of service (CoS), IP Security (IPsec), stateful firewalls, flow collection, and flow monitoring.
<i>System Log Messages Reference</i>	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
<b>J-Web User Guide</b>	
<i>J-Web Interface User Guide</i>	Describes how to use the J-Web graphical user interface (GUI) to configure, monitor, and manage Juniper Networks routing platforms.
<b>JUNOS API and Scripting Documentation</b>	
<i>JUNOScript API Guide</i>	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
<i>JUNOS XML API Configuration Reference</i>	Provides reference pages for the configuration tag elements in the JUNOS XML API.



**Table 3: Technical Documentation for Supported Routing Platforms (continued)**

Book	Description
<i>JUNOS XML API Operational Reference</i>	Provides reference pages for the operational tag elements in the JUNOS XML API.
<i>NETCONF API Guide</i>	Describes how to use the NETCONF API to monitor and configure Juniper Networks routing platforms.
<i>JUNOS Configuration and Diagnostic Automation Guide</i>	Describes how to use the commit script and self-diagnosis features of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts, how to use commit script macros to provide simplified aliases for frequently used configuration statements, and how to configure diagnostic event policies.
<b>Hardware Documentation</b>	
<i>Hardware Guide</i>	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
<i>PIC Guide</i>	Describes the routing platform's Physical Interface Cards (PICs). Each platform has its own PIC guide.
<i>DPC Guide</i>	Describes the Dense Port Concentrators (DPCs) for all MX-series routers.
<b>JUNOScope Documentation</b>	
<i>JUNOScope Software User Guide</i>	Describes the JUNOScope software graphical user interface (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.
<b>Advanced Insight Solutions (AIS) Documentation</b>	
<i>Advanced Insight Solutions Guide</i>	Describes the Advanced Insight Manager (AIM) application, which provides a gateway between JUNOS devices and Juniper Support Systems (JSS) for case management and intelligence updates. Explains how to run AI-Scripts on Juniper Networks devices.
<b>J-series Routing Platform Documentation</b>	
<i>Getting Started Guide</i>	Provides an overview, basic instructions, and specifications for J-series routing platforms. The guide explains how to prepare your site for installation, unpack and install the router and its components, install licenses, and establish basic connectivity. Use the <i>Getting Started Guide</i> for your router model.
<i>Basic LAN and WAN Access Configuration Guide</i>	Explains how to configure the interfaces on J-series Services Routers for basic IP routing with standard routing protocols, ISDN backup, and digital subscriber line (DSL) connections.
<i>Advanced WAN Access Configuration Guide</i>	Explains how to configure J-series Services Routers in virtual private networks (VPNs) and multicast networks, configure data link switching (DLSw) services, and apply routing techniques such as policies, stateless and stateful firewall filters, IP Security (IPsec) tunnels, and class-of-service (CoS) classification for safer, more efficient routing.

**Table 3: Technical Documentation for Supported Routing Platforms** (*continued*)

Book	Description
<i>Administration Guide</i>	Shows how to manage users and operations, monitor network performance, upgrade software, and diagnose common problems on J-series Services Routers.
<b>Release Notes</b>	
<i>JUNOS Release Notes</i>	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS, JUNOScript, and NETCONF manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
<i>Hardware Release Notes</i>	Describe the available documentation for the routing platform and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
<i>JUNOScope Release Notes</i>	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.
<i>AIS Release Notes</i>	Summarize AIS new features and guidelines, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide initial setup, upgrade, and downgrade procedures.
<i>AIS AI-Scripts Release Notes</i>	Summarize AI-Scripts new features, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide instructions for automatic and manual installation, including deleting and rolling back.
<i>J-series Services Router Release Notes</i>	Briefly describe Services Router features, identify known hardware problems, and provide upgrade and downgrade instructions.

**Table 4: JUNOS Software Network Operations Guides**

Book	Description
<i>Baseline</i>	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.
<i>Interfaces</i>	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
<i>MPLS</i>	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 routing platforms in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.

**Table 4: JUNOS Software Network Operations Guides** (*continued*)

Book	Description
<i>MPLS Log Reference</i>	Describes MPLS status and error messages that appear in the output of the <code>show mpls lsp extensive</code> 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.
<i>MPLS Fast Reroute</i>	Describes operational information helpful in monitoring and troubleshooting an MPLS network configured with fast reroute (FRR) and load balancing.
<i>Hardware</i>	Describes tasks for monitoring M-series and T-series routing platforms.

To configure and operate a J-series Services Router running JUNOS software with enhanced services, you must also use the configuration statements and operational mode commands documented in JUNOS configuration guides and command references. To configure and operate a WX Integrated Services Module, you must also use WX documentation.

**Table 5: JUNOS Software with Enhanced Services Documentation**

Book	Description
<b>All Platforms</b>	
<i>JUNOS Software Interfaces and Routing Configuration Guide</i>	Explains how to configure J-series interfaces for basic IP routing with standard routing protocols, ISDN service, firewall filters (access control lists), and class-of-service (CoS) traffic classification.
<i>JUNOS Software Security Configuration Guide</i>	Explains how to configure and manage security services such as stateful firewall policies, IP Security (IPsec) virtual private networks (VPNs), firewall screens, Network Address Translation (NAT), Public Key Cryptography, and Application Layer Gateways (ALGs).
<i>JUNOS Software Administration Guide</i>	Shows how to monitor J-series devices and routing operations, firewall and security services, system alarms and events, and network performance. This guide also shows how to administer user authentication and access, upgrade software, and diagnose common problems.
<i>JUNOS Software CLI Reference</i>	Provides the complete JUNOS software with enhanced services configuration hierarchy and describes the configuration statements and operational mode commands not documented in the standard JUNOS manuals.
<b>J-series Only</b>	
<i>JUNOS Software with Enhanced Services Design and Implementation Guide</i>	Provides guidelines and examples for designing and implementing IPsec VPNs, firewalls, and routing on J-series Services Routers running JUNOS software with enhanced services.

**Table 5: JUNOS Software with Enhanced Services Documentation (continued)**

Book	Description
<i>JUNOS Software with Enhanced Services Quick Start</i>	Explains how to quickly set up a J-series Services Router. This document contains router declarations of conformity.
<i>JUNOS Software with Enhanced Services J-series Services Router Hardware Guide</i>	Provides an overview, basic instructions, and specifications for J-series Services Routers. This guide explains how to prepare a site, unpack and install the router, replace router hardware, and establish basic router connectivity. This guide contains hardware descriptions and specifications.
<i>JUNOS Software with Enhanced Services Migration Guide</i>	Provides instructions for migrating an SSG device running ScreenOS software or a J-series Services Router running the JUNOS software to JUNOS software with enhanced services.
<i>WXC Integrated Services Module Installation and Configuration Guide</i>	Explains how to install and initially configure a WXC Integrated Services Module in a J-series Services Router for application acceleration.
<i>JUNOS Software with Enhanced Services Release Notes</i>	Summarizes new features and known problems for a particular release of JUNOS software with enhanced services on J-series Services Routers, including J-Web interface features and problems. The release notes also contain corrections and updates to the manuals and software upgrade and downgrade instructions for JUNOS software with enhanced services.

**Table 6: Additional Books Available Through <http://www.juniper.net/books>**

Book	Description
<i>Interdomain Multicast Routing</i>	Provides background and in-depth analysis of multicast routing using Protocol Independent Multicast sparse mode (PIM SM) and Multicast Source Discovery Protocol (MSDP); details any-source and source-specific multicast delivery models; explores multiprotocol BGP (MBGP) and multicast IS-IS; explains Internet Gateway Management Protocol (IGMP) versions 1, 2, and 3; lists packet formats for IGMP, PIM, and MSDP; and provides a complete glossary of multicast terms.
<i>JUNOS Cookbook</i>	Provides detailed examples of common JUNOS software configuration tasks, such as basic router configuration and file management, security and access control, logging, routing policy, firewalls, routing protocols, MPLS, and VPNs.
<i>MPLS-Enabled Applications</i>	Provides an overview of Multiprotocol Label Switching (MPLS) applications (such as Layer 3 virtual private networks [VPNs], Layer 2 VPNs, virtual private LAN service [VPLS], and pseudowires), explains how to apply MPLS, examines the scaling requirements of equipment at different points in the network, and covers the following topics: point-to-multipoint label switched paths (LSPs), DiffServ-aware traffic engineering, class of service, interdomain traffic engineering, path computation, route target filtering, multicast support for Layer 3 VPNs, and management and troubleshooting of MPLS networks.
<i>OSPF and IS-IS: Choosing an IGP for Large-Scale Networks</i>	Explores the full range of characteristics and capabilities for the two major link-state routing protocols: Open Shortest Path First (OSPF) and IS-IS. Explains architecture, packet types, and addressing; demonstrates how to improve scalability; shows how to design large-scale networks for maximum security and reliability; details protocol extensions for MPLS-based traffic engineering, IPv6, and multipoint-to-multipoint routing; and covers troubleshooting for OSPF and IS-IS networks.

**Table 6: Additional Books Available Through <http://www.juniper.net/books> (continued)**

Book	Description
<i>Routing Policy and Protocols for Multivendor IP Networks</i>	Provides a brief history of the Internet, explains IP addressing and routing (Routing Information Protocol [RIP], OSPF, IS-IS, and Border Gateway Protocol [BGP]), explores ISP peering and routing policies, and displays configurations for both Juniper Networks and other vendors' routers.
<i>The Complete IS-IS Protocol</i>	Provides the insight and practical solutions necessary to understand the IS-IS protocol and how it works by using a multivendor, real-world approach.

## Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net), or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. 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 (not required for *Network Operations Guides [NOGs]*)

## Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at <http://www.juniper.net/customers/support/downloads/710059.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

### Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>

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

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

### Opening a Case with JTAC

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

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

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

## **Part 1**

# **Network Management Introduction**

- Network Management Overview on page 3
- Complete Network Management Configuration Statements on page 7





## Chapter 1

# Network Management Overview

This chapter contains the following topic:

- Understanding the JUNOS Device Management Functions on page 3

### Understanding the JUNOS Device Management Functions

---

After you have installed the device into your network, you need to manage the device within your network. Device management can be divided into five tasks:

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

The JUNOS software network management features work in conjunction with an operations support system (OSS) to manage the devices within the network. The JUNOS software can assist you in performing these management tasks, as described in Table 7 on page 4.

**Table 7: JUNOS Device Management Features**

Task	JUNOS Software Feature
Fault management	<p>Monitor and see faults using:</p> <ul style="list-style-type: none"> <li>■ Operational mode commands—For more information on operational mode commands, see the <i>JUNOS System Basics and Services Command Reference</i>, <i>JUNOS Interfaces Command Reference</i>, and <i>JUNOS Routing Protocols and Policies Command Reference</i>.</li> <li>■ SNMP MIBs—For more information about SNMP MIBs, see “Juniper Networks Enterprise-Specific MIBs” on page 123.</li> <li>■ Standard SNMP traps—For more information about standard SNMP traps, see “Standard SNMP Traps” on page 143.</li> <li>■ Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.</li> <li>■ System log messages—For more information about how to configure system log messages, see the <i>JUNOS System Basics Configuration Guide</i>. For more information about how to view system log messages, see the <i>JUNOS System Log Messages Reference</i>.</li> </ul>
Configuration management	<ul style="list-style-type: none"> <li>■ Configure router attributes using the command-line interface (CLI), the JUNOScript API, and the NETCONF API. For more information on configuring the router using the CLI, see the <i>JUNOS System Basics Configuration Guide</i>. For more information on configuring the router using the APIs, see the <i>JUNOScript API Guide</i> and <i>NETCONF API Guide</i>.</li> <li>■ Configuration Management MIB—For more information about the Configuration Management MIB, see “Juniper Networks Enterprise-Specific MIBs” on page 123.</li> </ul>

**Table 7: JUNOS Device Management Features** *(continued)*

Task	JUNOS Software Feature
Accounting management	<p>Perform the following accounting-related tasks:</p> <ul style="list-style-type: none"> <li>■ 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 659.</li> <li>■ 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.</li> <li>■ Use per-ATM virtual circuit (VC) counters, available in the enterprise-specific ATM MIB.</li> <li>■ 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 123, “Configuring Class Usage Profiles” on page 675, the <i>JUNOS Network Interfaces Configuration Guide</i>, and the <i>JUNOS Policy Framework Configuration Guide</i>.</li> <li>■ Count packets as part of a firewall filter. For more information on firewall filter policies, see “Juniper Networks Enterprise-Specific MIBs” on page 123 and the <i>JUNOS Policy Framework Configuration Guide</i>.</li> <li>■ 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 <i>JUNOS Policy Framework Configuration Guide</i>.</li> </ul>
Performance management	<p>Monitor performance in the following ways:</p> <ul style="list-style-type: none"> <li>■ Use operational mode commands. For more information on monitoring performance using operational mode commands, see the <i>JUNOS System Basics and Services Command Reference</i>.</li> <li>■ Use firewall filter. For more information on performance monitoring using firewall filters, see the <i>JUNOS Policy Framework Configuration Guide</i>.</li> <li>■ 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 <i>JUNOS Policy Framework Configuration Guide</i>.</li> <li>■ Use the enterprise-specific Class-of-Service MIB. For more information on this MIB, see “Juniper Networks Enterprise-Specific MIBs” on page 123.</li> </ul>
Security management	<p>Assure security in your network in the following ways:</p> <ul style="list-style-type: none"> <li>■ Control access to the router and authenticate users. For more information on access control and user authentication, see the <i>JUNOS System Basics Configuration Guide</i>.</li> <li>■ Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see “Configuring the Local Engine ID” on page 49 and “Tracing SNMP Activity on a JUNOS Device” on page 46.</li> </ul>



## Chapter 2

# Complete Network Management Configuration Statements

This chapter contains the following topics:

- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

### Configuration Statements at the [edit accounting-options] Hierarchy Level

---

This topic shows all possible configuration statements at the [edit accounting-options] hierarchy level and 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 Hierarchy and RFC Reference*.

```
[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 {
    }
    files number;
    nonpersistent;
    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;
}
mib-profile profile-name {
  file filename;
  interval seconds;
  objects-names {
    mib-object-name;
  }
  operation operation-name;
}
routing-engine-profile profile-name {
  fields {
    field-name;
  }
  file filename;
  interval minutes;
}

```

## Configuration Statements at the [edit snmp] Hierarchy Level

---

This topic shows all possible configuration statements at the [edit snmp] hierarchy level and 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 Hierarchy and RFC Reference*.

```

[edit]
snmp {
  client-list client-list-name {
    ip-addresses;
  }
  community community-name {
    authorization authorization;
    client-list-name client-list-name;
    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;
        falling-threshold-interval seconds;
        interval seconds;
        request-type (get-next-request | get-request | walk-request);
        rising-event-index index;
        rising-threshold integer;
        sample-type type;
        startup-alarm alarm;
        syslog-subtag syslog-subtag;
        variable oid-variable;
    }
    event index {
        community community-name;
        description description;
        type type;
    }
}
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>
    <match regex>;
    flag flag;
}
trap-group group-name {
    categories {
        category;
    }
    destination-port port-number;
    routing-instance instance;
    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;
    routing-instance instance;
    tag-list tag-list;
    target-parameters target-parameters-name;
  }
  target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
      message-processing-model (v1 | v2c | v3);
      security-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**

# **Integrated Local Management Interface**

- Integrated Local Management Interface Overview on page 15



## Chapter 3

# Integrated Local Management Interface Overview

This chapter contains the following topic:

- Understanding Integrated Local Management Interface on page 15

## Understanding Integrated Local Management Interface

---

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. 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*) 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 with 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

- SNMP Overview on page 19
- Configuring SNMP on page 31
- SNMPv3 Overview on page 51
- Configuring SNMPv3 on page 53
- SNMP Remote Operations on page 87
- SNMP Support for Routing Instances on page 105
- Juniper Networks Enterprise-Specific MIBs on page 123
- Juniper Networks Enterprise-Specific SNMP Traps on page 131
- Standard SNMP Traps on page 143
- Summary of SNMP Configuration Statements on page 165
- Summary of SNMPv3 Configuration Statements on page 183





## Chapter 4

# SNMP Overview

This chapter contains the following topics:

- Understanding SNMP Implementation in the JUNOS Software on page 19
- Standard SNMP MIBs Supported by the JUNOS Software on page 22

### Understanding SNMP Implementation in the JUNOS Software

---

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

This topic covers the following sections:

- SNMP Architecture on page 19
- JUNOS SNMP Agent Features on page 22

### 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.

This topic contains the following sections:

- Management Information Base on page 20
- SNMP Traps and Informs on page 20

## 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 “Standard SNMP MIBs Supported by the JUNOS Software” on page 22.

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 123.

## 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 143.

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 131. For information

on system logging severity levels for SNMP traps, see “System Logging Severity Levels for SNMP Traps” on page 21.

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 76.

### **SNMP Trap Queuing**

The JUNOS software supports trap queuing to ensure that traps are not lost because of temporary unavailability of routes. Two types of queues, destination queues and a throttle queue, are formed to ensure delivery of traps and control the trap traffic.

The JUNOS software forms a destination queue when a trap to a particular destination is returned because the host is not reachable, and adds the subsequent traps to the same destination to the queue. The JUNOS software checks for availability of routes every 30 seconds, and sends the traps from the destination queue in a round-robin fashion. If the trap delivery fails, the trap is added back to the queue, and the delivery attempt counter and the next delivery attempt timer for the queue are reset. Subsequent attempts occur at progressive intervals of 1 minute, 2 minutes, 4 minutes, and 8 minutes. The maximum delay between the attempts is 8 minutes, and the maximum number of attempts is 10. After 10 unsuccessful attempts, the destination queue and all the traps in the queue are deleted.

The JUNOS software also has a throttle mechanism to control the number of traps (**throttle threshold**; default value of 100 traps) sent during a particular time period (**throttle interval**; default of 5 seconds) and to ensure consistency in trap traffic, especially when large number of traps are generated because of interface status changes. The throttle interval period begins when the first trap arrives at the throttle. All traps within the trap threshold are processed, and the traps beyond the threshold limit are queued. The maximum size of the throttle queue is 50k. When a trap is added to the throttle queue, or if the throttle queue has exceeded the maximum size, the trap is added back on top of the destination queue, and all subsequent attempts from the destination queue are stopped for a 30-second period, after which the destination queue restarts sending the traps.



**NOTE:** Users cannot configure the JUNOS software for trap queuing. Users cannot view any information about trap queues except what is available in the **syslog**.

---

### **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 143. For more information on system logging severity

levels for enterprise-specific traps, see “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.

## **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

## **Standard SNMP MIBs Supported by the JUNOS Software**

Table 8 on page 23 contains the list of standard SNMP MIBs and RFCs that are supported on various JUNOS platforms. RFCs can be found at <http://www.ietf.org>.



**NOTE:** In Table 8 on page 23, a value of 1 in any of the platform columns (M, T, J, MX, and EX) denotes that the corresponding MIB is supported on that particular platform, and a value of 0 denotes that the MIB is not supported on the platform.

**Table 8: Standard MIBs Supported on JUNOS Platforms**

MIB/RFC	Platforms				
	M	T	J	MX	EX
IEEE 802.1ab section 12.1, <i>Link Layer Discovery Protocol (LLDP) MIB</i>	0	0	0	0	1
IEEE, 802.3ad, <i>Aggregation of Multiple Link Segments</i>	1	1	1	1	1
Supported tables and objects:					
<ul style="list-style-type: none"> <li>■ dot3adAggPortTable, dot3adAggPortListTable, dot3adAggTable, and dot3adAggPortStatsTable</li> <li>■ dot3adAggPortDebugTable (only dot3adAggPortDebugRxState, dot3adAggPortDebugMuxState, dot3adAggPortDebugActorSyncTransitionCount, dot3adAggPortDebugPartnerSyncTransitionCount, dot3adAggPortDebugActorChangeCount, and dot3adAggPortDebugPartnerChangeCount)</li> <li>■ dot3adTablesLastChanged</li> </ul>					
<b>NOTE:</b> Gigabit Ethernet interfaces on J-series Services Routers do not support the 802.3ad MIB.					
RFC 1155, <i>Structure and Identification of Management Information for TCP/IP-based Internets</i>	1	1	1	1	1
RFC 1157, <i>A Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1
RFC 1195, <i>Use of OSI IS-IS for Routing in TCP/IP and Dual Environments</i> (only the objects isisSystem, isisMANAreaAddr, isisAreaAddr, isisSysProtSupp, isisSummAddr, isisCirc, isisCircLevel, isisPacketCount, isisISAdj, isisISAdjAreaAddr, isisAdjIPAddr, isisISAdjProtSupp, isisRa, and isisIPRA are supported)	1	1	1	1	0
RFC 1212, <i>Concise MIB Definitions</i>	1	1	1	1	0
RFC 1213, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> . The JUNOS software supports the following areas:	1	1	1	1	1
<ul style="list-style-type: none"> <li>■ MIB II and its SNMP version 2 derivatives, including: <ul style="list-style-type: none"> <li>■ Statistics counters</li> <li>■ IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, <i>IP Forwarding Table MIB</i>)</li> <li>■ SNMP management</li> <li>■ Interface management</li> </ul> </li> <li>■ SNMPv1 Get, GetNext requests, and version 2 GetBulk request</li> <li>■ JUNOS software-specific secured access list</li> <li>■ Master configuration keywords</li> <li>■ Reconfigurations upon SIGHUP</li> </ul>					

**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
RFC 1215, <i>A Convention for Defining Traps for use with the SNMP</i> (only MIB II SNMP version 1 traps and version 2 notifications)	1	1	1	1	1
RFC 1406, <i>Definitions of Managed Objects for the DS1 and E1 Interface Types</i> (T1 MIB is supported)	1	1	1	0	0
RFC 1407, <i>Definitions of Managed Objects for the DS3/E3 Interface Type</i> (T3 MIB is supported)	1	1	1	0	0
RFC 1657, <i>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2</i>	1	1	1	1	1
RFC 1695, <i>Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2</i>	1	1	1	0	0
RFC 1850, <i>OSPF Version 2 Management Information Base</i> (except for the <code>ospfOriginateNewLsas</code> and <code>ospfRxNewLsas</code> objects, the Host Table, and the traps <code>ospfOriginateLSA</code> , <code>ospfLsdbOverflow</code> , and <code>ospfLsdbApproachingOverflow</code> )	1	1	1	1	1
RFC 1901, <i>Introduction to Community-based SNMPv2</i>	1	1	1	1	0
RFC 1905, <i>Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)</i>	1	1	1	1	1
RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> (replaced by RFC 3418)	1	1	1	1	1
RFC 2011, <i>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</i>	1	1	1	1	1
RFC 2012, <i>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</i>	1	1	1	1	1
RFC 2013, <i>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</i>	1	1	1	1	1
RFC 2024, <i>Definitions of Managed Objects for Data Link Switching Using SMIv2</i> (except for the <code>dlswInterface</code> and <code>dlswSdlc</code> object groups; the <code>dlswDirLocateMacTable</code> , <code>dlswDirNBTable</code> , and <code>dlswDirLocateNBTable</code> tables; the <code>dlswCircuitDiscReasonLocal</code> and <code>dlswCircuitDiscReasonRemote</code> tabular objects; and the <code>dlswDirMacCacheNextIndex</code> and <code>dlswDirNBCacheNextIndex</code> scalar objects; read-only access)	1	1	1	1	0
RFC 2096, <i>IP Forwarding Table MIB</i> (The <code>ipCidrRouteTable</code> has been extended to include the tunnel name when the next hop is through an RSVP-signaled LSP.)	1	1	1	1	1
RFC 2115, <i>Management Information Base for Frame Relay DTEs Using SMIv2</i> ( <code>frDlcmiTable</code> only; <code>frCircuitTable</code> and <code>frErrTable</code> are not supported.)	1	1	1	1	0

**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
RFC 2233, <i>The Interfaces Group MIB Using SMIv2</i>	1	1	1	1	0
RFC 2287, <i>Definitions of System-Level Managed Objects for Applications</i> (only the objects <code>sysAppInstallPkgTable</code> , <code>sysAppInstallElmtTable</code> , <code>sysAppElmtRunTable</code> , and <code>sysAppMapTable</code> )	1	1	1	1	1
RFC 2465, <i>Management Information Base for IP Version 6: Textual Conventions and General Group</i> (except for IPv6 interface statistics)	1	1	1	1	0
RFC 2495, <i>Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types</i> (except for <code>dsx1FarEndConfigTable</code> , <code>dsx1FarEndCurrentTable</code> , <code>dsx1FarEndIntervalTable</code> , <code>dsx1FarEndTotalTable</code> , and <code>dsx1FracTable</code> )	1	1	1	0	0
RFC 3896, <i>Definitions of Managed Objects for the DS3/E3 Interface Type</i> (except <code>dsx3FarEndConfigTable</code> , <code>dsx3FarEndCurrentTable</code> , <code>dsx3FarEndIntervalTable</code> , <code>dsx3FarEndTotalTable</code> , and <code>dsx3FracTable</code> )	1	1	1	0	0
RFC 2515, <i>Definitions of Managed Objects for ATM Management</i> (except <code>atmVpCrossConnectTable</code> , <code>atmVcCrossConnectTable</code> , and <code>aal5VccTable</code> )	1	1	1	0	0
RFC 3592, <i>Definitions of Managed Objects for the SONET/SDH Interface Type</i>	1	1	1	0	0
RFC 2570, <i>Introduction to Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1
RFC 2571, <i>An Architecture for Describing SNMP Management Frameworks</i> (read-only access)	1	1	1	1	1
RFC 2572, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> (read-only access)	1	1	1	1	1
RFC 2576, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1
RFC 2578, <i>Structure of Management Information Version 2 (SMIv2)</i>	1	1	1	1	1
RFC 2579, <i>Textual Conventions for SMIv2</i>	1	1	1	1	1
RFC 2580, <i>Conformance Statements for SMIv2</i>	1	1	1	1	0
RFC 2662, <i>Definitions of Managed Objects for ADSL Lines</i> (J-series Services Routers. All MIB tables, objects, and traps are applicable for the ADSL ATU-R agent.)	1	1	1	1	0
RFC 2665, <i>Definitions of Managed Objects for the Ethernet-like Interface Types</i>	1	1	1	1	1

**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> (except row creation, the <b>Set</b> operation, and the object <b>vrpStatsPacketLengthErrors</b> )	1	1	1	1	1
RFC 2790, <i>Host Resources MIB</i>	1	1	1	1	1
<ul style="list-style-type: none"> <li>Only the <b>hrStorageTable</b>. The file systems <b>/</b>, <b>/config</b>, <b>/var</b>, and <b>/tmp</b> always return the same index number. When SNMP restarts, the index numbers for the remaining file systems might change.</li> <li>Only the objects of the <b>hrSystem</b> and <b>hrSWInstalled</b> groups.</li> </ul>					
RFC 2819, <i>Remote Network Monitoring Management Information Base</i> (the <b>etherStatsTable</b> for Ethernet interfaces only and the objects <b>alarmTable</b> , <b>eventTable</b> , and <b>logTable</b> )	1	1	1	1	1
RFC 2863, <i>The Interfaces Group MIB</i>	1	1	1	1	1
RFC 2864, <i>The Inverted Stack Table Extension to the Interfaces Group MIB</i>	1	1	1	1	0
RFC 2922, <i>The Physical Topology (PTOPO) MIB</i>	0	0	0	0	1
RFC 2925, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i> (only the objects <b>pingCtlTable</b> , <b>pingResultsTable</b> , <b>pingProbeHistoryTable</b> , <b>pingMaxConcurrentRequests</b> , <b>traceRouteCtlTable</b> , <b>traceRouteResultsTable</b> , <b>traceRouteProbeHistoryTable</b> , and <b>traceRouteHopsTable</b> )	1	1	1	1	1
RFC 2932, <i>IPv4 Multicast Routing MIB</i>	1	1	1	1	1
RFC 2933, <i>Internet Group Management Protocol (IGMP) MIB</i>	1	1	1	1	0
RFC 2934, <i>Protocol Independent Multicast MIB for IPv4</i>	1	1	1	1	0
RFC 2981, <i>Event MIB</i>	1	1	1	1	0
RFC 3014, <i>Notification Log MIB</i>	1	1	1	1	0
RFC 3109, <i>IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol</i>	1	1	1	1	0
RFC 3411, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i>	1	1	1	1	0
RFC 3412, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	0
RFC 3413, <i>Simple Network Management Protocol (SNMP) Applications</i> (except for the proxy MIB)	1	1	1	1	1



**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
RFC 3414, <i>User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)</i>	1	1	1	1	1
RFC 3415, <i>View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1
RFC 3416, <i>Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	0
RFC 3417, <i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	0
RFC 3418, <i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i> (replaces RFC 1907)	1	1	1	1	0
RFC 3498, <i>Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures</i> (implemented under the Juniper Networks enterprise branch [jnxExperiment])	1	1	1	0	0
RFC 3592, <i>Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type</i>	1	1	1	1	0
RFC 3621, <i>Power Ethernet MIB</i>	0	0	0	0	1
RFC 3637, <i>Definitions of Managed Objects for the Ethernet WAN Interface Sublayer</i> (except <code>etherWisDeviceTable</code> , <code>etherWisSectionCurrentTable</code> , and <code>etherWisFarEndPathCurrentTable</code> )	1	1	1	1	0
RFC 3811, <i>Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management</i>	1	1	1	1	0
RFC 3812, <i>Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)</i> (read only access)	1	1	1	1	0
<ul style="list-style-type: none"> <li>■ MPLS tunnels as interfaces are not supported.</li> <li>■ The following objects in the <code>TunnelResource</code> table are not supported: <code>mplsTunnelResourceMeanRate</code>, <code>mplsTunnelResourceMaxBurstSize</code>, <code>mplsTunnelResourceMeanBurstSize</code>, <code>mplsTunnelResourceExBurstSize</code>, <code>mplsTunnelResourceWeight</code>.</li> <li>■ <code>mplsTunnelPerfTable</code> and <code>mplsTunnelCRLDPResTable</code> are not supported.</li> <li>■ <code>mplsTunnelCHopTable</code> supported on ingress routers only.</li> </ul>					
<p><b>NOTE:</b> The branch used by the proprietary LDP MIB (<code>ldpmib.mib</code>) conflicts with RFC 3812. <code>ldpmib.mib</code> has been deprecated and replaced by <code>jnx-mpls-ldp.mib</code>.</p>					

**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
RFC 3813, <i>Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)</i> (read only access. <code>mplsInterfacePerfTable</code> , <code>mplsInSegmentPerfTable</code> , <code>mplsOutSegmentPerfTable</code> , <code>mplsInSegmentMapTable</code> , <code>mplsXCUp</code> , and <code>mplsXCDown</code> are not supported.)	1	1	1	1	0
RFC 3815, <i>Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)</i> (only <code>mplsLdpLsrId</code> and <code>mplsLdpSesPeerAddrTable</code> )	1	1	1	1	0
RFC 3826, <i>The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model</i>	1	1	1	1	0
RFC 4188, <i>Definitions of Managed Objects for Bridges</i> —Supports 802.1D STP(1998). Supports only the following subtrees and objects: <ul style="list-style-type: none"> <li>■ <code>dot1dStp</code> subtree is supported on MX-series Ethernet Services routers.</li> <li>■ <code>dot1dTpFdbAddress</code>, <code>dot1dTpFdbPort</code>, and <code>dot1dTpFdbStatus</code> objects from the <code>dot1dTpFdbTable</code> of the <code>dot1dTp</code> subtree are supported on EX-series Ethernet switches.</li> </ul> <b>NOTE:</b> <code>dot1dTpLearnedEntryDiscards</code> and <code>dot1dTpAgingTime</code> objects are supported on M and T series routers.	0	0	0	1	1
RFC 4318, <i>Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol</i> —Supports 802.1w and 802.1t extensions for RSTP.	1	1	1	1	0
RFC 4363b <i>Q-Bridge VLAN MIB</i>	0	0	0	0	1
RFC 4801, <i>Definitions of Textual Conventions for Generalized Multiprotocol Label Switching (GMPLS) Management Information Base (MIB)</i> (read-only access)	1	1	1	1	0
RFC 4802, <i>Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering (TE) Management Information Base (MIB)</i> (read-only access. <code>gmplsTunnelReversePerfTable</code> , <code>gmplsTeScalars</code> , <code>gmplsTunnelTable</code> , <code>gmplsTunnelARHopTable</code> , <code>gmplsTunnelCHopTable</code> , and <code>gmplsTunnelErrorTable</code> are not supported.)	1	1	1	1	0
RFC 4803, <i>Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) Management Information Base (MIB)</i> (read-only access. <code>gmplsLabelTable</code> and <code>gmplsOutsegmentTable</code> are not supported.)	1	1	1	1	0
<b>NOTE:</b> The tables in GMPLS TE (RFC 4802) and LSR (RFC 4803) MIBs are extensions of the corresponding tables from the MPLS TE (RFC 3812) and LSR (RFC 3813) MIBs and use the same index as the MPLS MIB tables.					

**Table 8: Standard MIBs Supported on JUNOS Platforms** (continued)

MIB/RFC	Platforms				
	M	T	J	MX	EX
Internet Assigned Numbers Authority, <i>IANAiftype Textual Convention MIB</i> (referenced by RFC 2233, available at <a href="ftp://ftp.isi.edu/mib/ianaiftype.mib">ftp://ftp.isi.edu/mib/ianaiftype.mib</a> )	1	1	1	1	1
Internet draft draft-blumenthal-aes-usm-08.txt, <i>The AES Cipher Algorithm in the SNMP User-based Security Model</i>	1	1	1	1	0
Internet draft draft-ietf-atommib-sonetaps-mib-10.txt, <i>Definitions of Managed Objects for SONET Linear APS Architectures</i> (as defined under the Juniper Networks enterprise branch [jnxExperiment] only)	1	1	1	1	0
Internet draft draft-ietf-bfd-mib-02.txt, <i>Bidirectional Forwarding Detection Management Information Base</i> (Represented by mib-jnx-bfd-exp.txt and implemented under the Juniper Networks enterprise branch [jnxExperiment]. Read only. Includes bfdSessUp and bfdSessDown traps. Does not support bfdSessPerfTable and bfdSessMapTable.)	1	1	1	1	1
Internet draft draft-ietf-idmr-igmp-mib-13.txt, <i>Internet Group Management Protocol (IGMP) MIB</i>	1	1	1	1	0
Internet draft draft-ietf-idr-bgp4-mibv2-04.txt, <i>Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version</i> (only jnxBgpM2PrefixInPrefixes, jnxBgpM2PrefixInPrefixesAccepted, and jnxBgpM2PrefixInPrefixesRejected objects)	1	1	1	1	0
Internet draft draft-reeder-snmpv3-usm-3desede-00.txt, <i>Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode</i>	1	1	1	1	0
Internet draft draft-ietf-isis-wg-mib-07.txt, <i>Management Information Base for IS-IS</i> , (only isisISAdjTable, isisISAdjAreaAddrTable, isisISAdjIPAddrTable, and isisISAdjProtSuppTable)	1	1	1	1	1
Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt, <i>MPLS/BGP Virtual Private Network Management Information Base Using SMIv2</i> (only mplsVpnScalars, mplsVpnVrfTable, mplsVpnPerTable, and mplsVpnVrfRouteTargetTable)	1	1	1	1	0
Internet draft draft-ietf-msdp-mib-07.txt, <i>Multicast Source Discovery protocol MIB</i> (except msdpEstablished, msdpBackwardTransition, and msdpRequestsTable)	1	1	1	1	0
Internet draft draft-ietf-ospf-ospfv3-mib-11.txt, <i>Management Information Base for OSPFv3</i> (Represented by mib-jnx-ospfv3mib.txt and implemented under the Juniper Networks enterprise branch [jnxExperiment]. Support for ospfv3NbrTable only. Read only. Object names are prefixed by jnx. For example, jnxOspf3NbrTable, jnxOspf3NbrAddressType, and jnxOspf3NbrPriority.)	1	1	1	1	0

**Table 8: Standard MIBs Supported on JUNOS Platforms** (*continued*)

MIB/RFC	Platforms				
	M	T	J	MX	EX
Internet draft draft-ietf-idmr-pim-mib-09.txt, <i>Protocol Independent Multicast (PIM) MIB</i>	1	1	1	1	0
ESO Consortium MIB, which can be found at <a href="http://www.snmp.com/eso/">http://www.snmp.com/eso/</a>	1	1	1	1	0

## Chapter 5

# Configuring SNMP

This chapter contains the following topics:

- Configuring SNMP on a JUNOS Device on page 32
- Configuring the System Contact on a JUNOS Device on page 34
- Configuring the System Location for a JUNOS Device on page 34
- Configuring the System Description on a JUNOS Device on page 34
- Filtering Duplicate SNMP Requests on page 35
- Configuring the Commit Delay Timer on page 35
- Configuring the System Name on page 35
- Configuring the SNMP Community String on page 36
- Adding a Group of Clients to an SNMP Community on page 37
- Configuring SNMP Trap Options and Groups on a JUNOS Device on page 38
- Configuring SNMP Trap Options on page 39
- Configuring SNMP Trap Groups on page 41
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 44
- Configuring MIB Views on page 45
- Tracing SNMP Activity on a JUNOS Device on page 46
- Configuring the Local Engine ID on page 49

## Configuring SNMP on a JUNOS Device

---

By default, Simple Network Management Protocol (SNMP) is disabled on JUNOS devices. To enable SNMP on a JUNOS device, you must include the SNMP configuration statements at the `[edit snmp]` hierarchy level.

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.

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

```
snmp {
  client-list client-list-name {
    ip-addresses;
  }
  community community-name {
    authorization authorization;
    client-list-name client-list-name;
    clients {
      address restrict;
    }
    routing-instance routing-instance-name {
      clients {
        addresses;
      }
    }
    logical-system logical-system-name {
      routing-instance routing-instance-name {
        clients {
          addresses;
        }
      }
    }
  }
  view view-name;
}
contact contact;
description description;
engine-id {
  (local engine-id | use-mac-address | use-default-ip-address);
}
filter-duplicates;
health-monitor {
  falling-threshold integer;
  interval seconds;
  rising-threshold integer;
}
interface [ interface-names ];
```

```

location location;
name name;
nonvolatile {
    commit-delay seconds;
}
rmon {
    alarm index {
        description text-description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        falling-threshold-interval seconds;
        request-type (get-next-request | get-request | walk-request);
        sample-type type;
        startup-alarm alarm;
        syslog-subtag syslog-subtag;
        variable oid-variable;
    }
    event index {
        community community-name;
        description text-description;
        type type;
    }
}
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>
        <match regex>;
    flag flag;
}
trap-group group-name {
    categories {
        category;
    }
    destination-port port-number;
    routing-instance instance;
    targets {
        address;
    }
    version (all | v1 | v2);
}
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
view view-name {
    oid object-identifier (include | exclude);
}
}

```

## Configuring the System Contact on a JUNOS Device

---

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 for a JUNOS Device

---

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 on a JUNOS Device

---

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 on JUNOS devices. 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 a JUNOS device 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 CLI User Guide*.

## Configuring the System Name

---

The JUNOS software enables you to override the system name by including 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 in a JUNOS configuration, include the community statement at the [edit snmp] hierarchy level:

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

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

The default authorization level for a community is **read-only**. To allow **Set** requests within a community, you need to define that community as **authorization read-write**. For **Set** requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the **view** statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information on the **view** statement, see "Configuring MIB Views" on page 45.

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.
    }
  }
}
```

## Adding a Group of Clients to an SNMP Community

---

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

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

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

You can configure a prefix list at the `[edit policy options]` hierarchy level. Support for prefix lists in the SNMP community configuration enables you to use a single list to configure the SNMP and routing policies. For more information on the `prefix-list` statement, see the *JUNOS Policy Framework Configuration Guide*.

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

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



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

---

**Example: Defining a Client List**

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

**Example: Adding a Client List to an SNMP Community**

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

**Example: Adding a Prefix List to an SNMP Community**

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

## Configuring SNMP Trap Options and Groups on a JUNOS Device

---

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

## 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 41.

This topic contains the following sections:

- Configuring the Source Address for SNMP Traps on page 40
- Configuring the Agent Address for SNMP Traps on page 41

## 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;
    }
  }
}
```

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

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

For each trap group that you define, you must include the **target** statement to define at least one system as the recipient of the SNMP traps in the trap group. Specify the IPv4 or IPv6 address of each recipient, not its 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 143 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.

Specify the routing instance used by the trap group in the **routing-instance** statement. All targets configured in the trap group use this routing instance.

A trap group can receive the following categories:

- **authentication**—Authentication failures
- **chassis**—Chassis or 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.

- **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 alarm indication signal (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**—Virtual tributary (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**—Virtual Router Redundancy Protocol (VRRP) events such as new-master or authentication failures

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

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**.

**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 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 123.

## Tracing SNMP Activity on a JUNOS Device

---

Simple Network Management Protocol (SNMP) tracing operations track activity for SNMP agents and record the information in log files. The logged error descriptions provide detailed information to help you solve problems faster.

By default, the JUNOS software does not trace any SNMP activity. If you include the `traceoptions` statement at the `[edit snmp]` hierarchy level, the default tracing behavior is the following:

- Important activities are logged in files located in the `/var/log` directory. Each log is named after the SNMP agent that generates it. Currently, the following log files are created in the `/var/log` directory when the `traceoptions` statement is used:
  - `chassisd`
  - `craftd`
  - `ilmid`
  - `mib2d`
  - `rmopd`
  - `serviced`
  - `snmpd`
- When a trace file named *filename* reaches its maximum size, it is renamed *filename.0*, then *filename.1*, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. (For more information about how log files are created, see the *JUNOS System Log Messages Reference*.)
- Log files can be accessed only by the user who configures the tracing operation.

You cannot change the directory (`/var/log`) in which trace files are located. However, you can customize the other trace file settings by including the following statements at the `[edit snmp]` hierarchy level:

```
[edit snmp]
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>
    <match regex>;
  flag flag;
}
```

These statements are described in the following sections:

- Configuring the SNMP Log Filename on page 47
- Configuring the Number and Size of SNMP Log Files on page 47
- Configuring Access to the Log File on page 47
- Configuring a Regular Expression for Lines to Be Logged on page 47
- Configuring the Trace Operations on page 48

## Configuring the SNMP Log Filename

By default, the name of the file that records trace output is `snmpd`. You can specify a different name by including the `file` statement at the `[edit snmp traceoptions]` hierarchy level:

```
[edit snmp traceoptions]
file filename;
```

## Configuring the Number and Size of SNMP Log Files

By default, when the trace file reaches 128 kilobytes (KB) in size, it is renamed `filename.0`, then `filename.1`, and so on, until there are three trace files. Then the oldest trace file (`filename.2`) is overwritten.

You can configure the limits on the number and size of trace files by including the following statements at the `[edit snmp traceoptions]` hierarchy level:

```
[edit snmp traceoptions]
file files number size size;
```

For example, set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation (`filename`) reaches 2 MB, `filename` is renamed `filename.0`, and a new file called `filename` is created. When the new `filename` reaches 2 MB, `filename.0` is renamed `filename.1` and `filename` is renamed `filename.0`. This process repeats until there are 20 trace files. Then the oldest file (`filename.19`) is overwritten by the newest file (`filename.0`).

The number of files can be from 2 through 1000 files. The file size of each file can be from 10 KB through 1 gigabyte (GB).

## Configuring Access to the Log File

By default, log files can be accessed only by the user who configures the tracing operation.

To specify that any user can read all log files, include the `file world-readable` statement at the `[edit snmp traceoptions]` hierarchy level:

```
[edit snmp traceoptions]
file world-readable;
```

To explicitly set the default behavior, include the `file no-world-readable` statement at the `[edit snmp traceoptions]` hierarchy level:

```
[edit snmp traceoptions]
file no-world-readable;
```

## Configuring a Regular Expression for Lines to Be Logged

By default, the trace operation output includes all lines relevant to the logged activities.

You can refine the output by including the `match` statement at the `[edit snmp traceoptions file filename]` hierarchy level and specifying a regular expression (regex) to be matched:

```
[edit snmp traceoptions]
file filename match regex;
```

## Configuring the Trace Operations

By default, only important activities are logged. You can specify which trace operations are to be logged by including the following `flag` statement (with one or more tracing flags) at the `[edit snmp traceoptions]` hierarchy level:

```
[edit snmp traceoptions]
flag {
  all;
  configuration;
  database;
  events;
  general;
  interface-stats;
  nonvolatile-sets;
  pdu;
  policy;
  protocol-timeouts;
  routing-socket;
  server;
  subagent;
  timer;
  varbind-error;
}
```

Table 9 on page 48 describes the meaning of the SNMP tracing flags.

**Table 9: SNMP Tracing Flags**

Flag	Description	Default Setting
all	Log all operations.	Off
configuration	Log reading of configuration at the <code>[edit snmp]</code> hierarchy level.	Off
database	Log events involving storage and retrieval in events database.	Off
events	Log important events.	Off
general	Log general events.	Off
interface-stats	Log physical and logical interface statistics.	Off
nonvolatile-set	Log nonvolatile SNMP set request handling.	Off
pdu	Log SNMP request and response packets.	Off

**Table 9: SNMP Tracing Flags** (*continued*)

Flag	Description	Default Setting
policy	Log policy processing.	Off
protocol-timeouts	Log SNMP response timeouts.	Off
routing-socket	Log routing socket calls.	Off
server	Log communication with processes that are generating events.	Off
subagent	Log subagent restarts.	Off
timer	Log internal timer events.	Off
varbind-error	Log variable binding errors.	Off

To display the end of the log for an agent, issue the `show log agentd | last` operational mode command:

```
[edit]
user@host# run show log agentd | last
```

where *agent* is the name of an SNMP agent.

#### Example: Tracing SNMP Activity

Trace information about 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 56.





## Chapter 6

# SNMPv3 Overview

This chapter contains the following topic:

- SNMPv3 Overview on page 51

## SNMPv3 Overview

---

In contrast to SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2), SNMP version 3 (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 57
- Configuring MIB Views on page 45

- Defining Access Privileges for an SNMP Group on page 62
- Configuring SNMPv3 Traps on a JUNOS Device on page 68
- Configuring SNMP Informs on page 76

## Chapter 7

# Configuring SNMPv3

This chapter contains the following topics:

- Complete SNMPv3 Configuration Statements on page 54
- Minimum SNMPv3 Configuration on a JUNOS Device on page 55
- Configuring the Local Engine ID on page 56
- Creating SNMPv3 Users on page 57
- Configuring the SNMPv3 Authentication Type on page 58
- Configuring the Encryption Type on page 59
- Example: Creating SNMPv3 Users Configuration on page 61
- Defining Access Privileges for an SNMP Group on page 62
- Configuring the Access Privileges Granted to a Group on page 63
- Example: Access Privilege Configuration on page 65
- Assigning Security Names to Groups on page 66
- Example: Security Group Configuration on page 68
- Configuring SNMPv3 Traps on a JUNOS Device on page 68
- Configuring the SNMPv3 Trap Notification on page 69
- Configuring the Trap Notification Filter on page 70
- Configuring the Trap Target Address on page 70
- Defining and Configuring the Trap Target Parameters on page 74
- Configuring SNMP Informs on page 76
- Configuring the Remote Engine and Remote User on page 77
- Example: Configuring the Remote Engine ID and Remote Users on page 78
- Configuring the Inform Notification Type and Target Address on page 78
- Example: Configuring the Inform Notification Type and Target Address on page 80
- Configuring the SNMPv3 Community on page 80
- Example: SNMPv3 Community Configuration on page 82
- Example: SNMPv3 Configuration on page 82

## Complete SNMPv3 Configuration Statements

---

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;
  routing-instance instance;
  tag-list tag-list;
  target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
  notify-filter profile-name;
  parameters {
    message-processing-model (v1 | v2c | v3);
    security-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;
        }
    }
}
}
}

```

## Minimum SNMPv3 Configuration on a JUNOS Device

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

```

[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;

```

## Configuring the SNMPv3 Authentication Type

---

By default, in a JUNOS configuration the SNMPv3 authentication type is set to none.

This topic includes the following sections:

- Configuring MD5 Authentication on page 58
- Configuring SHA Authentication on page 58
- Configuring No Authentication on page 59

### Configuring 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 eight characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring 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 eight 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 topic includes the following sections:

- Configuring the Advanced Encryption Standard Algorithm on page 59
- Configuring the Data Encryption Algorithm on page 60
- Configuring Triple DES on page 60
- Configuring No Encryption on page 60

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

## Defining Access Privileges for an SNMP Group

The Simple Network Management Protocol version 3 (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, 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 45.

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;
    }
}

```

## Configuring the Access Privileges Granted to a Group

---

This topics includes the following sections:

- Configuring the Group on page 63
- Configuring the Security Model on page 63
- Configuring the Security Level on page 63
- Associating MIB Views with an SNMP User Group on page 64

### 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 vacm 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-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;
  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 45.

---

This section describes the following topics related to this configuration:

- Configuring the Notify View on page 65
- Configuring the Read View on page 65
- Configuring the Write View on page 65

## 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 ww2;
      }
    }
  }
}
group group3 {
  default-context-prefix {
    security-model v1 {          #Define an SNMPv3 security model
      security-level none {
        read-view rv3;
        write-view ww3;
      }
    }
  }
}
}

```

## 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 topic includes the following sections:

- Configuring the Security Model on page 66
- Configuring the Security Name on page 67
- Configuring the Group on page 67

### 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 57. For information about configuring a community string, see “Configuring the SNMPv3 Community” on page 80.



**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 62.

## 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 SNMPv3 Traps on a JUNOS Device

---

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 76.

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;
```

```

routing-instance instance;
tag-list tag-list;
target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
  notify-filter profile-name;
  parameters {
    message-processing-model (v1 | v2c | v3);
    security-model (usm | v1 | v2c);
    security-level (authentication | none | privacy);
    security-name security-name;
  }
}

```

## Configuring the SNMPv3 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`.

### 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 system (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;
routing-instance instance;
tag-list tag-list;
target-parameters target-parameters-name;
```

This section includes the following topics:

- Configuring the Address on page 71
- Configuring the Address Mask on page 71
- Configuring the Port on page 72
- Configuring the Routing Instance on page 72
- Configuring the Tag List on page 72
- Applying Target Parameters on page 73

## 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 defines a range of addresses. For information about how to configure the community string, see “Configuring the SNMPv3 Community” on page 80.

## Configuring the Port

By default, the UDP port is set to 162. To configure a different port number, 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-number* is the SNMP target port number.

## Configuring the Routing Instance

Traps are sent over the default routing instance. To configure the routing instance for sending traps, include the `routing-instance` statement at the `[edit snmp v3 target-address target-address-name]` hierarchy level:

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

*instance* is the name of the routing instance. To configure a routing instance within a logical system, specify the logical system name followed by the routing instance name. Use a slash ( / ) to separate the two names (for example, `test-lr/test-ri`). To configure the default routing instance on a logical system, specify the logical system name followed by `default` (for example, `test-lr/default`).

## 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 as a space-separated list enclosed within double quotes.

For information about how to specify a tag at the `[edit snmp v3 notify notify-name]` hierarchy level, see “Configuring the SNMPv3 Trap Notification” on page 69.

### 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 list
    target-parameters tp3;
  }
```



**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 and Configuring 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 topic includes the following sections:

- Applying the Trap Notification Filter on page 74
- Configuring the Target Parameters on page 74

### 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 70.

### 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 on page 75
- Configuring the Security Model on page 75
- Configuring the Security Level on page 75
- Configuring the Security Name on page 76

### 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-model` 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 the notification is generated. If the v1 or v2c security models are used, **security-name** identifies the SNMP community used when the notification is generated.



**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.

## Configuring SNMP Informs

The 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 one of these conditions occurs:

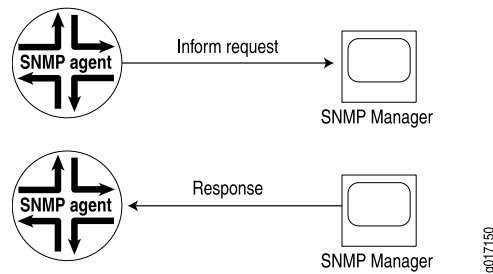
- The receiver (target) of the inform returns an acknowledgment to the SNMP agent.
- A specified number of unsuccessful retransmissions have been attempted and the agent discards the inform message.

If the sender never receives a response, the inform can be sent again. Thus, informs are more likely to reach their intended destination than traps are. Informs use the

same communications channel as traps (same socket and port) but have different protocol data unit (PDU) types.

Informs are more reliable than traps, but they consume more network and router resources (See Figure 1 on page 77). Unlike a trap, an inform is held in memory until a response is received or the timeout is reached. Also, traps are sent only once, whereas an inform may be retried several times. Use informs when it is important that the SNMP manager receive all notifications. However, if you are more concerned about network traffic or router memory, use traps.

**Figure 1: Inform Request and Response**



For information on configuring SNMP traps, see “Configuring SNMPv3 Traps on a JUNOS Device” on page 68.

## 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 "$9$D0jP536901Riktu1lcSwY2gUj5QF3
/CYgQF/Cu0xN-bwgZGiqP5iH.5TF/9WLX7wYoaUkqfoaAp
OBEhSreW87s24aUjsY4ZDjq.RhcyWLNdbg4Zs
YJDHkTQ69Apu1EcyrWQF/tuOREYg4ajHmPQF39
Ygz3n6At8XxNYgik.PTz7-ikmfn6vW8XVw";
      }
    }
    privacy-des {
      privacy-key "$9$MZZXxdwYgJUjIKGiH5T69Au0IrlM7NbeK24
aJDjO1IRylM8Xbwg1R24aJDjHqm5n/Ap0ORhn6evLXbwmf5T
/CRhSyKM5QEcleW87-Vbs4JGD.mT-VwgaZkqfTznAphSrlM8yr
Wx7dsYTzF36Atu01EcpuNdwYoa69CuRhcyM8rlaZGjq.01IEhr";
    }
  }
}

```

## 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;
    routing-instance instance;
    tag-list tag-list;
    target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
        message-processing-model (v1 | v2c | v3);
        security-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 72.

**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 acknowledgment. If no acknowledgment 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 acknowledgment is received. The default is 3. If no acknowledgment 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 SNMP notifications are generated. Informs require a v3 message processing model.

**security-model** defines the security model to use when SNMP notifications are generated. 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 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 SNMPv3 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 on page 81
- Configuring the Security Names on page 81
- Configuring the Tag on page 82

## 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 access control is set up. 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 you configure 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: SNMPv3 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; # Applies 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 to send
} # In this case, includes all traps
notify-filter nf2 {
    oid 1.3.6.1.4.1 include; # Sends enterprise-specific traps only
}
notify-filter nf3 {
    oid 1.3.6.1.2.1.1.5 include; # Sends 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 allowed to be used with
}
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; # Applies 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 { # Defines the target parameters
    notify-filter nf1; # Specifies 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 { #Defines authentication and encryption for SNMPv3 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;
    }
  }
}
vacm {

```

```

access {
  group san-francisco { #Defines the access privileges for the group
    default-context-prefix { # called 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; # called san-francisco
    }
    security-name bob {
      group new-york;
    }
    security-name elizabeth {
      group chicago;
    }
  }
}

```



## Chapter 8

# SNMP Remote Operations

This chapter contains the following topics:

- SNMP Remote Operations Overview on page 87
- Using the Ping MIB on page 90
- Starting a Ping Test on page 90
- Monitoring a Running Ping Test on page 91
- Gathering Ping Test Results on page 94
- Stopping a Ping Test on page 96
- Interpreting Ping Variables on page 96
- Using the Traceroute MIB on page 97

### SNMP Remote Operations Overview

---

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 123.

This topic covers the following sections:

- SNMP Remote Operation Requirements on page 88
- Setting SNMP Views on page 88
- Setting Trap Notification for Remote Operations on page 89
- Using Variable-Length String Indexes on page 89
- Enabling Logging on page 90

## 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.

## 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 36 and `community`.

For more information on the `view` statement, see “Configuring MIB Views” on page 45 and `view`.

## 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 group-name]` hierarchy level:

```
[edit snmp trap-group group-name]
  categories {
    category;
  }
  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 41.

## 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 process 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 a JUNOS Device” on page 46.

If the remote operations process 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.

## 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 88.

There are two ways to start a ping test:

- Using Multiple Set PDUs on page 91
- Using a Single Set PDU on page 91

### ***Using Multiple Set PDUs***

You can use multiple `Set` request PDUs (multiple PDUs, with one or more varbinds 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 acknowledgment of the SNMP `Set` request is sent back to the client:

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

For more information, see the following sections:

- `pingResultsTable` on page 92
- `pingProbeHistoryTable` on page 93
- `Generating Traps` on page 94

## ***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 the `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 occur:

- `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 93.

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 41 and “Example: Setting Trap Notification for Remote Operations” on page 89.

## Gathering Ping Test Results

You can either poll `pingResultsOperStatus` to find out when the test is complete or request that a trap be sent when the test is complete. For more information on `pingResultsOperStatus`, see “`pingResultsTable`” on page 92. For more information on Ping MIB traps, see “Generating Traps” on page 94.

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 contains probes like those in Table 10 on page 95.

**Table 10: 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 11 on page 95.

**Table 11: 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 12 on page 96.

**Table 12: 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 on page 97
- Monitoring a Running Traceroute Test on page 98
- Monitoring Traceroute Test Completion on page 102
- Gathering Traceroute Test Results on page 103
- Stopping a Traceroute Test on page 104
- Traceroute Variables on page 104

### 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`:

- `traceRouteCtlOwnerIndexSnmAdminString`
- `traceRouteCtlTestNameSnmAdminString`
- `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 88.

There are two ways to start a traceroute test:

- Using Multiple Set PDUs on page 98
- Using a Single Set PDU on page 98

### Using Multiple Set PDUs

You can use multiple **Set** request PDUs (multiple PDUs, with one or more varbinds 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 acknowledgment of the SNMP **Set** request is sent back to the client:

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

For more information, see the following sections:

- `traceRouteResultsTable` on page 98
- `traceRouteProbeResultsTable` on page 99
- `traceRouteHopsTable` on page 101
- Generating Traps on page 102

### 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.

The `traceRouteCtlProbesPerHop` number of probes is 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 41 and “Example: Setting Trap Notification for Remote Operations” on page 89.

## 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 that a trap be sent when the test is complete. For more information on `traceResultsOperStatus`, see “`traceRouteResultsTable`” on page 98. For more information on Traceroute MIB traps, see “Generating Traps” on page 102.

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 eight hops to the target (the target being number eight).
- 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 13 on page 103.

**Table 13: `traceRouteProbeHistoryTable`**

HistoryIndex	HistoryHopIndex	HistoryProbeIndex
31	7	1
32	7	2
33	7	3

**Table 13: traceRouteProbeHistoryTable** (continued)

HistoryIndex	HistoryHopIndex	HistoryProbeIndex
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

# SNMP Support for Routing Instances

This chapter contains the following topics:

- Understanding SNMP Support for Routing Instances on page 105
- Support Classes for MIB Objects on page 106
- Identifying a Routing Instance on page 107
- Enabling SNMP Access over Routing Instances on page 108
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 108
- Example: Configuring Interface Settings for a Routing Instance on page 109
- Configuring Access Lists for SNMP Access over Routing Instances on page 110
- Trap Support for Routing Instances on page 111
- MIB Support Details on page 111

## Understanding SNMP Support for Routing Instances

---

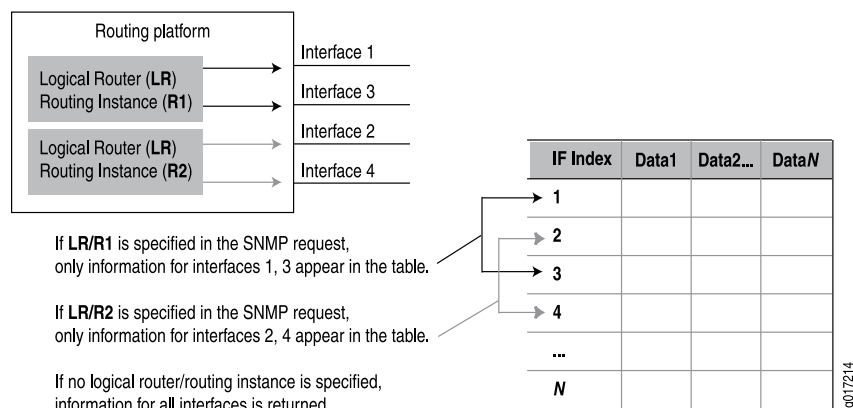
The JUNOS software enables SNMP managers for all routing instances to request and manage SNMP data related to the corresponding routing instances and logical system networks.

In the JUNOS software:

- Clients from routing instances other than the default can access MIB objects and perform SNMP operations only on the logical system networks to which they belong.
- Clients from the default routing instance can access information related to all routing instances and logical system networks.

Before JUNOS Release 8.4, only the SNMP manager in the default routing instance (`inet.0`) had access to the MIB objects

With the increase in virtual private network (VPN) service offerings, this feature is useful particularly for service providers who need to obtain SNMP data for specific routing instances (see Figure 2 on page 106). Service providers can use this information for their own management needs or export the data for use by their customers.

**Figure 2: SNMP Data for Routing Instances**

If no routing instance is specified in the request, the SNMP agent operates as before:

- For non-routing table objects, all instances will be exposed.
- For routing table objects, only those associated with the default routing instance will be exposed.



**NOTE:** The actual protocol data units (PDUs) are still exchanged over the default (inet.0) routing instance, but the data contents returned are dictated by the routing instance specified in the request PDUs.

## Support Classes for MIB Objects

When a routing instance is specified, all routing-related MIB objects return data maintained by the routing instance in the request. For all other MIB objects, the data returned is segregated according to that routing instance. For example, only those interfaces assigned to that routing instance (for example, the logical interfaces [ifls] as well as their corresponding physical interfaces [ifds]) are exposed by the SNMP agent. Similarly, objects with an unambiguous attachment to an interface (for example, **addresses**) are segregated as well.

For those objects where the attachment is ambiguous (for example, objects in **sysAppMIB**), no segregation is done and all instances are visible in all cases.

Another category of objects is visible only when no logical system is specified (only within the default logical system) regardless of the routing instance within the default logical system. Objects in this category are Chassis MIB objects, objects in the SNMP group, RMON alarm, event and log groups, Ping MIB objects, configuration management objects, and V3 objects.

In summary, to support routing instances, MIB objects fall into one of the following categories:



- Class 1—Data is segregated according to the routing instance in the request. This is the most granular of the segregation classes.
- Class 2—Data is segregated according to the logical system specified in the request. The same data is returned for all routing instances that belong to a particular logical system. Typically, this applies to routing table objects where it is difficult to extract routing instance information or where routing instances do not apply.
- Class 3—Data is exposed only for the default logical system. The same set of data is returned for all routing instances that belong to the default logical system. If you specify another logical system (not the default), no data is returned. Typically this class applies to objects implemented in subagents that do not monitor logical system changes and register their objects using only the default context (for example, Chassis MIB objects).
- Class 4—Data is not segregated by routing instance. The same data is returned for all routing instances. Typically, this applies to objects implemented in subagents that monitor logical system changes and register or deregister all their objects for each logical system change. Objects whose values cannot be segregated by routing instance fall into this class.

See “MIB Support Details” on page 111 for a list of the objects associated with each class.

## Identifying a Routing Instance

---

With this feature, routing instances are identified by either the context field in v3 requests or encoded in the community string in v1 or v2c requests.

When encoded in a community string, the routing instance name appears first and is separated from the actual community string by the @ character.

To avoid conflicts with valid community strings that contain the @ character, the community is parsed only if typical community string processing fails. For example, if a routing instance named **RI** is configured, an SNMP request with **RI@public** is processed within the context of the **RI** routing instance. Access control (views, source address restrictions, access privileges, and so on) is applied according to the actual community string (the set of data after the @ character—in this case **public**). However, if the community string **RI@public** is configured, the PDU is processed according to that community and the embedded routing instance name is ignored.

Logical systems perform a subset of the actions of a physical router and have their own unique routing tables, interfaces, policies, and routing instances. When a routing instance is defined within a logical system, the logical system name must be encoded along with the routing instance using a slash ( / ) to separate the two. For example, if the routing instance **RI** is configured within the logical system **LS**, that routing instance must be encoded within a community string as **LS/RI@public**. When a routing instance is configured outside a logical system (within the default logical system), no logical system name (or / character) is needed.

Also, when a logical system is created, a default routing instance (named **default**) is always created within the logical system. This name should be used when querying

data for that routing instance (for example, `LS/default@public`). For v3 requests, the name *logical system/routing instance* should be identified directly in the context field.



**NOTE:** To identify a VLAN spanning tree instance (VSTP on MX series Ethernet Services router), specify the routing instance name followed by a double colon (::) and the VLAN ID. For example, to identify VSTP instance for VLAN 10 in the global default routing instance, include `default::10@public` in the `context` (SNMPv3) or `community` (SNMPv1 or v2) string.

## Enabling SNMP Access over Routing Instances

To enable SNMP managers in routing instances other than the default routing instance to access SNMP information, include the `routing-instance-access` statement in the SNMP configuration.

```
[edit]
user@router1# show snmp
routing-instance-access;
```

If this statement is not included in the SNMP configuration, the JUNOS software will not allow SNMP managers from routing instances other than the default routing instance to access SNMP information.

## Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community

You can specify the routing instance along with the client information when you add a client to an SNMP community. To specify the routing instance to which a client belongs, include the `routing-instance` statement followed by the routing instance name and client information in the SNMP configuration.

The following example shows the configuration statement to add routing instance `test-ri` to SNMP community `community1`.



**NOTE:** Routing instances specified at the `[edit snmp community community-name]` hierarchy level are added to the default logical system in the community.

```
[edit snmp]
community community1 {
  clients {
    10.209.152.33/32;
  }
  routing-instance test-ri {
    clients {
      10.19.19.1/32;
    }
  }
}
```

If the routing instance is defined within a logical system, include the `routing-instance` statement at the `[edit snmp community community-name logical-system logical-system-name]` hierarchy level, as in the following example:

```
[edit snmp]
community community1 {
  clients {
    10.209.152.33/32;
  }
  logical-system test-LS {
    routing-instance test-ri {
      clients {
        10.19.19.1/32;
      }
    }
  }
}
```

### Example: Configuring Interface Settings for a Routing Instance

This example shows an 802.3ad ae0 interface configuration allocated to a routing instance named INFRtd:

```
[edit chassis]
aggregated-devices {
  ethernet {
    device-count 5;
  }
}
[edit interfaces ae0]
vlan-tagging;
aggregated-ether-options {
  minimum-links 2;
  link-speed 100m;
}
unit 0 {
  vlan-id 100;
  family inet {
    address 10.1.0.1/24;
  }
}
[edit interfaces fe-1/1/0]
fastether-options {
  802.3ad ae0;
}
[edit interfaces fe-1/1/1]
fastether-options {
  802.3ad ae0;
}
[edit routing-instances]
INFRtd {
  instance-type virtual-router;
  interface fe-1/1/0.0;
  interface fe-1/1/1.0;
  interface fe-1/1/5.0;
```

```

interface ae0.0;
protocols {
  ospf {
    area 0.0.0.0 {
      interface all;
    }
  }
}
}

```

The following `snmpwalk` command shows how to retrieve SNMP-related information from `router1` and the `802.3ae` bundle interface belonging to routing instance `INFrtd` with the SNMP community `public`:

```

freebsd# snmpwalk -Os router1 INFrtd@public dot3adAggTable
dot3adAggMACAddress.59 = 0:90:69:92:93:f0
dot3adAggMACAddress.65 = 0:90:69:92:93:f0
dot3adAggActorSystemPriority.59 = 0
dot3adAggActorSystemPriority.65 = 0
dot3adAggActorSystemID.59 = 0:0:0:0:0:0
dot3adAggActorSystemID.65 = 0:0:0:0:0:0
dot3adAggAggregateOrIndividual.59 = true(1)
dot3adAggAggregateOrIndividual.65 = true(1)
dot3adAggActorAdminKey.59 = 0
dot3adAggActorAdminKey.65 = 0
dot3adAggActorOperKey.59 = 0
dot3adAggActorOperKey.65 = 0
dot3adAggPartnerSystemID.59 = 0:0:0:0:0:0
dot3adAggPartnerSystemID.65 = 0:0:0:0:0:0
dot3adAggPartnerSystemPriority.59 = 0
dot3adAggPartnerSystemPriority.65 = 0
dot3adAggPartnerOperKey.59 = 0
dot3adAggPartnerOperKey.65 = 0
dot3adAggCollectorMaxDelay.59 = 0
dot3adAggCollectorMaxDelay.65 = 0

```

## Configuring Access Lists for SNMP Access over Routing Instances

You can create and maintain access lists to manage access to SNMP information. Access list configuration enables you to allow or deny SNMP access to clients of a specific routing instance.

The following example shows how to create an access list:

```

[edit snmp]
routing-instance-access {
  access-list {
    ri1 restrict;
    ls1/default;
    ls1/ri2;
    ls1*;
  }
}

```

The configuration given in the example:

- Restricts clients in `ri1` from accessing SNMP information.
- Allows clients in `ls1/default`, `ls1/ri2`, and all other routing instances with names starting with `ls1` to access SNMP information.

You can use the wildcard character (\*) to represent a string in the routing instance name.



**NOTE:** You cannot restrict the SNMP manager of the default routing instance from accessing SNMP information.

## Trap Support for Routing Instances

When configured under the `trap-group` object, all v1 and v2c traps that apply to routing instances (or interfaces belonging to a routing instance) have the routing instance name encoded in the community string. The encoding is identical to that used in request PDUs.

For traps configured under the v3 framework, the routing instance name is carried in the context field when the v3 message processing model has been configured. For other message processing models (v1 or v2c), the routing instance name is not carried in the trap message header (and not encoded in the community string).

You can restrict the trap receivers from receiving traps that are not related to the logical system networks to which they belong. To do this, include the `logical-system-trap-filter` statement at the `[edit snmp]` hierarchy level:

```
[edit snmp]
logical-system-trap-filter;
```

If the `logical-system-trap-filter` statement is not included in the SNMP configuration, all traps are forwarded to the configured routing instance destinations. However, even when this statement is configured, the trap receiver associated with the default routing instance will receive all SNMP traps.

## MIB Support Details

Table 14 on page 111 shows enterprise-specific MIB objects supported by the JUNOS software and provides notes detailing how they are handled when a routing instance is specified in an SNMP request. An en dash (–) indicates that the item is not applicable.

**Table 14: MIB Support for Routing Instances (Juniper Networks MIBs)**

Object	Support Class	Description/Notes
jnxProducts(1)	–	Product Object IDs

**Table 14: MIB Support for Routing Instances (Juniper Networks MIBs)** *(continued)*

Object	Support Class	Description/Notes
jnxServices(2)	–	Services
jnxMibs(3)	Class 3	Objects will be exposed only for the default logical system.
jnxBoxAnatomy(1)		
mpls(2)	Class 2	All instances within a logical system will be exposed. Data will not be segregated down to the routing instance level.
ifJnx(3)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxAlarms(4)	Class 3	Objects will be exposed only for the default logical system.
jnxFirewalls(5)	Class 4	Data is not segregated by routing instance. All instances will be exposed.
jnxDCUs(6)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxPingMIB(7)	Class 3	Objects will be exposed only for the default logical system.
jnxTraceRouteMIB(8)	Class 3	Objects will be exposed only for the default logical system.
jnxATM(10)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxIpv6(11)	Class 4	Data is not segregated by routing instance. All instances will be exposed.
jnxIpv4(12)	Class 1	jnxIpv4AddrTable(1). Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxRmon(13)	Class 3	jnxRmonAlarmTable(1). Objects will be exposed only for the default logical system.
jnxLdp(14)	Class 2	jnxLdpTrapVars(1). All instances within a logical system will be exposed. Data will not be segregated down to the routing instance level.

**Table 14: MIB Support for Routing Instances (Juniper Networks MIBs)** *(continued)*

Object	Support Class	Description/Notes
jnxCos(15)  jnxCosIfqStatsTable(1) jnxCosFcTable(2) jnxCosFcidTable(3) jnxCosQstatTable(4)	Class 3	Objects will be exposed only for the default logical system.
jnxScu(16)  jnxScuStatsTable(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxRpf(17)  jnxRpfStatsTable(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxCfgMgmt(18)	Class 3	Objects will be exposed only for the default logical system.
jnxPMon(19)  jnxPMonFlowTable(1)  jnxPMonErrorTable(2)  jnxPMonMemoryTable(3)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxSonet(20)  jnxSonetAlarmTable(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxAtmCos(21)  jnxCosAtmVcTable(1)  jnxCosAtmScTable(2)  jnxCosAtmVcQstatsTable(3)  jnxCosAtmTrunkTable(4)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
ipSecFlowMonitorMIB(22)	–	–
jnxMac(23)  jnxMacStats(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
apsMIB(24)	Class 3	Objects will be exposed only for the default logical system.
jnxChassisDefines(25)	Class 3	Objects will be exposed only for the default logical system.

**Table 14: MIB Support for Routing Instances (Juniper Networks MIBs)** *(continued)*

Object	Support Class	Description/Notes
jnxVpnMIB(26)	Class 2	All instances within a logical system will be exposed. Data will not be segregated down to the routing instance level.
jnxSericesInfoMib(27)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxCollectorMIB(28)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.
jnxHistory(29)	–	–
jnxSpMIB(32)	Class 3	Objects will be exposed only for the default logical system.

Table 15 on page 115 shows Class 1 MIB objects (standard and enterprise-specific MIBs) supported by the JUNOS software. With Class 1 objects, only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance will be exposed.



**Table 15: Class 1 MIB Objects (Standard and Juniper MIBs)**

Class	MIB	Objects
Class 1	802.3ad.mib	(dot2adAgg) MIB objects:
		dot3addAggTable
		dot3adAggTablw
		dot3adAggPortListTable
		dot3adAggPortTable
		dot3adAggPortStatsTable
		dot3adAggPortDebugTable
	rfc2863a.mib	ifTable
		ifXTable
		ifStackTable
	rfc2011a.mib	ipAddrTable
		ipNetToMediaTable
	rtmib.mib	ipForward (ipCidrRouteTable)
	rfc2665a.mib	dot3StatsTable
		dot3ControlTable
		dot3PauseTable
	rfc2495a.mib	dsx1ConfigTable
		dsx1CurrentTable
		dsx1IntervalTable
		dsx1TotalTable
		dsx1FarEndCurrentTable
		dsx1FarEndIntervalTable
		dsx1FarEndTotalTable
		dsx1FracTable ...
	rfc2496a.mib	dsx3 (dsx3ConfigTable)
	rfc2115a.mib	frDlcmiTable (and related MIB objects)
	rfc3592.mib	

**Table 15: Class 1 MIB Objects (Standard and Juniper MIBs)** *(continued)*

Class	MIB	Objects
		sonetMediumTable (and related MIB objects)
	rfc3020.mib	mfrMIB
		mfrBundleTable
		mfrMibBundleLinkObjects
		mfrBundleIfIndexMappingTable
		(and related MIB objects)
	ospf2mib.mib	All objects
	ospf2trap.mib	All objects
	bgpmib.mib	All objects
	rfc2819a.mib	Example: etherStatsTable

**Table 15: Class 1 MIB Objects (Standard and Juniper MIBs)** *(continued)*

Class	MIB	Objects
Class 1	rfc2863a.mib	Examples:  ifXtable  ifStackTable
	rfc2665a.mib	etherMIB
	rfc2515a.mib	atmMIB objects  Examples:  atmInterfaceConfTable  atmVplTable  atmVclTable
	rfc2465.mib	ip-v6mib  Examples:  ipv6IfTable  ipv6AddrPrefixTable  ipv6NetToMediaTable  ipv6RouteTable
	rfc2787a.mib	vrp mib
	rfc2932.mib	ipMRouteMIB  ipMRouteStdMIB
	mroutemib.mib	ipMRoute1MIBObjects
	isismib.mib	isisMIB
	pimmib.mib	pimMIB
	msdpmib.mib	msdpmib
	jnx-if-extensions.mib	Examples:  ifJnxTable  ifChassisTable
	jnx-dcu.mib	jnxDCUs
	jnx-atm.mib	

**Table 15: Class 1 MIB Objects (Standard and Juniper MIBs)** *(continued)*

Class	MIB	Objects
		Examples:
		jnxAtmIfTable
		jnxAtmVcTable
		jnxAtmVpTable
	jnx-ipv4.mib	jnxipv4
	jnx-cos.mib	Example: jnxIpv4AddrTable
		Examples:
		jnxCosIfqStatsTable
	jnx-scu.mib	jnxCosQstatTable
		Example: jnxScuStatsTable
Class 1	jnx-rpf.mib	Example: jnxRpfStatsTable
	jnx-pmon.mib	Example: jnxPMonFlowTable
	jnx-sonet.mib	Example: jnxSonetAlarmTable
	jnx-atm-cos.mib	Examples:
		jnxCosAtmVcTable
		jnxCosAtmVcScTable
		jnxCosAtmVcQstatsTable
		jnxCosAtmTrunkTable
	jnx-mac.mib	Example: jnxMacStatsTable
	jnx-services.mib	Example: jnxSvcFlowTableAggStatsTable
	jnx-coll.mib	jnxCollectorMIB
		Examples:
		jnxCollPicIfTable
		jnxCollFileEntry

Table 16 on page 119 shows Class 2 MIB objects (standard and enterprise-specific MIBs) supported by the JUNOS software. With Class 2 objects, all instances within a logical system will be exposed. Data will not be segregated down to the routing instance level.

**Table 16: Class 2 MIB Objects (Standard and Juniper MIBs)**

Class	MIB	Objects
Class 2	rfc3813.mib	mplsLsrStdMIB  Examples:  mplsInterfaceTable  mplsInSegmentTable  mplsOutSegmentTable  mplsLabelStackTable  mplsXCTable  (and related MIB objects)
	igmpmib.mib	igmpStdMIB
	l3vpn.mib	mplsVpnMIB
	jnx-mpls.mib	Example: mplsLspList
	jnx-ldp.mib	jnxLdp  Example: jnxLdpStatsTable
	jnx-vpn.mib	jnxVpnMIB
	jnx-bgpmib2.mib	jnxBgpM2Experiment

Table 17 on page 120 shows Class 3 MIB objects (standard and enterprise-specific MIBs) supported by the JUNOS software. With Class 3, objects will be exposed only for the default logical system.

**Table 17: Class 3 MIB Objects (Standard and Juniper MIBs)**

Class	MIB	Objects
Class 3	rfc2819a.mib	rmonEvents
		alarmTable
		logTable
		eventTable
		agentxMIB
	rfc2925a.mib	pingmib
	rfc2925b.mib	tracerouteMIB
	jnxchassis.mib	jnxBoxAnatomy
	jnx-chassis-alarm.mib	jnxAlarms
	jnx-ping.mib	jnxPingMIB
	jnx-traceroute.mib	jnxTraceRouteMIB
	jnx-rmon.mib	jnxRmonAlarmTable
	jnx-cos.mib	Example: jnxCosFcTable
	jnx-cfgmgmt.mib	Example: jnxCfgMgmt
	jnx-sonetaps.mib	apsMIBObjects
	jnx-sp.mib	jnxSpMIB
	ggsn.mib	ejnmobileipABmib
	rfc1907.mib	snmpModules
	snmpModules	Examples:
		snmpMIB snmpFrameworkMIB

Table 18 on page 121 shows Class 4 MIB objects (standard and enterprise-specific MIBs) supported by the JUNOS software. With Class 4 objects, data is not segregated by routing instance. All instances will be exposed.

**Table 18: Class 4 MIB Objects (Standard and Juniper MIBs)**

Class	MIB	Objects
Class 4	system	Example: sysORTable
	rfc2011a.mib	ip (ipDefaultTTL, ipInReceives) icmp
	rfc2012a.mib	tcp tcpConnTable ipv6TcpConnTable
	rfc2013a.mib	udp udpTable ipv6UdpTable
	rfc2790a.mib	hrSystem
	rfc2287a.mib	sysAppIObj
	jnx-firewall.mib	jnxFirewalls
	jnx-ipv6.mib	jnxIpv6





## Chapter 10

# Juniper Networks Enterprise-Specific MIBs

This chapter contains the following section

- Juniper Networks Enterprise-Specific MIBs on page 123

## Juniper Networks Enterprise-Specific MIBs

---

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



**NOTE:** For detailed interpretation of Juniper Networks enterprise-specific MIBs, see Part 7, “Juniper Networks Enterprise-Specific MIBs” on page 287.

---

- AAA Objects MIB—Provides support for monitoring user authentication, authorization, and accounting through the RADIUS, LDAP, SecurID, and local authentication servers. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-user-aaa.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-user-aaa.txt).
- Access Authentication Objects MIB—Provides support for monitoring firewall authentication, including data about the users trying to access firewall-protected resources and the firewall authentication service itself. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-auth.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-auth.txt).
- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-alarm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-alarm.txt).
- Analyzer MIB—Contains analyzer and remote analyzer data related to port mirroring on the EX-series Ethernet switches. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-analyzer.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-analyzer.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/junos92/swconfig-net-mgmt/mib-jnx-atm-cos.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-atm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-atm.txt).
- BFD MIB—Provides support for monitoring Bidirectional Forwarding Detection (BFD) sessions. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bfd.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bfd.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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bgpmib2.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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, whereas the Chassis Definitions for Router Model MIB provides information that changes less often. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt).
- Chassis Forwarding MIB—Enables J-series Services Routers to fully support the JUNOS health monitor. This MIB extends the scope of health monitoring to include JUNOS forwarding process (fwdd) components. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-fwdd.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-fwdd.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/junos92/swconfig-net-mgmt/mib-jnx-cos.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-cfgmgmt.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dcu.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dcu.txt).

- **DNS Objects MIB**—Provides support for monitoring DNS proxy queries, requests, responses, and failures. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-dns.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-dns.txt).
- **Dynamic Flow Capture MIB**—Provides support for monitoring the operational status of dynamic flow capture PICs. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dfc.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dfc.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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mac.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mac.txt).
- **Event MIB**—Defines a generic trap that can be generated using an op script or event policy. This MIB provides the ability to specify a system log string and raise a trap if that system log string is found. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-event.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-event.txt).
- **Experimental MIB**—Contains object identifiers for experimental MIBs. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-exp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-firewall.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-coll.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-hostresources.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-hostresources.txt).
- **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/junos92/swconfig-net-mgmt/mib-jnx-if-extensions.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-if-extensions.txt).
- **IP Forward MIB**—Extends the standard IP Forwarding Table MIB (RFC 2096) to include CIDR forwarding information. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipforward.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipforward.txt).
- **IPSec Monitoring MIB**—Provides operational and statistical information related to the IPSec and IKE tunnels on Juniper Networks routing platforms. For a

downloadable version of this MIB, see

[www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-monitor-asp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-monitor-asp.txt).

- IPsec Generic Flow Monitoring Object MIB—Based on `jnx-ipsec-monitor-mib`, this MIB provides support for monitoring IPsec and IPsec VPN management objects. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-flow-mon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-flow-mon.txt).
- IPsec VPN Objects MIB—Provides support for monitoring IPsec and IPsec VPN management objects for Juniper security product lines. This MIB is an extension of `jnx-ipsec-flow-mon.mib`. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-ipsec-vpn.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-ipsec-vpn.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/junos92/swconfig-net-mgmt/mib-jnx-ipv4.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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/junos92/swconfig-net-mgmt/mib-jnx-ipv6.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipv6.txt).
- L2ALD MIB—Contains information on Layer-2 Address Learning Daemon and related traps, such as routing instance MAC limit trap and interface MAC limit trap. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2ald.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2ald.txt).
- L2CP Features MIB—Provides information about Layer 2 Control Protocols-based features on MX-series Ethernet Services routers. Currently, the JUNOS software supports only the `jnxDot1dStpPortRootProtectEnabled`, `jnxDot1dStpPortRootProtectState`, and `jnxPortRootProtectStateChangeTrap` objects. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2cp-features.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2cp-features.txt).
- L2TP MIB—Provides information on Layer 2 Transport Protocol (L2TP) tunnels and sessions. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2tp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2tp.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/junos92/swconfig-net-mgmt/mib-jnx-ldp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ldp.txt).
- Multiple Instance Multiple Spanning Tree protocol (MIMSTP) MIB—Provides information on MSTP instances (that is, routing instances of type Virtual Switch/Layer 2 control, also known as virtual contexts), MSTIs within the MSTP instance, and VLANs associated with the MSTI. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mimstp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mimstp.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/junos92/swconfig-net-mgmt/mib-jnx-mpls.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls.txt).



**NOTE:** To collect information about MPLS statistics on transit routers, use the enterprise-specific RSVP MIB (`mib-jnx-rsvp.txt`) instead of the enterprise-specific MPLS MIB (`mib-jnx-mpls.txt`).

- MPLS LDP MIB—Contains object definitions as described in RFC 3815, *Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)*. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls-ldp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls-ldp.txt).



**NOTE:** Objects in the MPLS LDP MIB were supported in earlier releases of JUNOS software as a proprietary LDP MIB (`mib-ldpmib.txt`). Because the branch used by the proprietary LDP (`mib-ldpmib.txt`) conflicts with RFC 3812, the proprietary LDP MIB (`mib-ldpmib.txt`) has been deprecated and replaced by the enterprise-specific MPLS LDP MIB (`mib-jnx-mpls-ldp.txt`).

- Network Address Translation (NAT) Objects MIB—Provides support for monitoring network address translation (NAT). This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-nat.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-nat.txt).
- Packet Forwarding Engine MIB—Provides notification statistics for Packet Forwarding Engines. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pfe.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pfe.txt).
- PAE Extension MIB—Extends the standard IEEE802.1x PAE Extension MIB, and contains information for Static MAC Authentication. The enterprise-specific PAE Extension MIB is supported only on EX-series Ethernet switches. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pae-extension.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pae-extension.txt).
- Passive Monitoring MIB—Performs traffic flow monitoring and lawful interception of packets transiting between two routers. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pmon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ping.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ping.txt).
- Policy Objects MIB—Provides support for monitoring the security policies that control the flow of traffic from one zone to another. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-policy.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-policy.txt).
- Real-time Performance Monitoring Protocol (RPM) MIB—Provides real-time performance-related data and enables you to access jitter measurements and calculations via SNMP. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpm.txt).

- Reverse-Path-Forwarding MIB—Monitors statistics for traffic that is rejected because of reverse-path-forwarding (RPF) processing. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpf.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpf.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/junos92/swconfig-net-mgmt/mib-jnx-rsvp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rsvp.txt).



**NOTE:** To collect information about MPLS statistics on transit routers, use the enterprise-specific RSVP MIB ([mib-jnx-rsvp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rsvp.txt)) instead of the enterprise-specific MPLS MIB ([mib-jnx-mpls.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls.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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rmon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rmon.txt).
- Secure Access Port MIB—Contains information about secure access port configuration on EX-series Ethernet switches. The EX-series Ethernet switches use DHCP snooping and dynamic ARP inspection mechanisms to extend security capabilities on interfaces. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-secure-access-port.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-secure-access-port.txt).
- Security Interface Extension Objects MIB—Provides support for the security management of interfaces. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-if-ext.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-if-ext.txt).
- Security Screening Objects MIB—Defines the MIB for the Juniper Networks Enterprise Firewall screen functionality. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-screening.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-screening.txt).
- Services PIC MIB—Provides statistics for Adaptive Services (AS) PICs and defines notifications for AS PICs. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sp.txt).
- SONET/SDH Interface Management MIB—Monitors the current alarm for each SONET/SDH interface. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sonet.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sonet.txt).
- SONET Automatic Protection Switching MIB—Monitors any SONET interface that participates in Automatic Protection Switching (APS). For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sonetaps.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 a downloadable version of this MIB, see

[www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-scu.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-scu.txt).

- Structure of Management Information MIB—Explains how the Juniper Networks enterprise-specific MIBs are structured. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-smi.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-smi.txt).
- Structure of Management Information MIB for EX-series Ethernet switches—Defines a MIB branch for switching-related MIB definitions for the EX-series Ethernet switches. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ex-smi.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ex-smi.txt).
- Structure of Management Information MIB—Contains object identifiers (OIDs) for the security branch of the MIBs used in the JUNOS software with enhanced services product, services and traps. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-smi.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-smi.txt).
- System Log MIB—Enables notification of an SNMP trap-based application when an important system log message occurs. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-syslog.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-syslog.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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-traceroute.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-traceroute.txt).
- Utility MIB—Provides SNMP support for exposing JUNOS data and has tables that contain information on each type of data, such as integer and string. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-util.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-util.txt).
- Virtual Chassis MIB—Contains information about virtual chassis on EX-series Ethernet switches. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-virtual-chassis.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-virtual-chassis.txt).
- VLAN MIB—Contains information about prestandard IEEE 802.10 VLANs and their association with LAN emulation clients. The enterprise-specific VLAN MIB is supported only on EX-series Ethernet switches. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vlan.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vlan.txt).
- VPN MIB—Provides monitoring for Layer 3 VPNs, Layer 2 VPNs, and virtual private LAN service (VPLS) (read access only). For a downloadable version of the MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vpn.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vpn.txt).
- VPN Certificate Objects MIB—Provides support for monitoring the local and CA certificates loaded on the router. This MIB is currently supported only by JUNOS software with enhanced services. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-cert.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-cert.txt).





## Chapter 11

# 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 on page 131
- Juniper Networks Enterprise-Specific SNMP Version 2 Traps on page 135
- Juniper Networks Enterprise-Specific LDP Traps on page 139
- Disabling LDP Traps on page 139
- Juniper Networks Enterprise-Specific Version 2 Traps on EX-Series Ethernet Switches on page 139
- Juniper Networks Enterprise-Specific Version 2 Traps on MX960 Platforms on page 139
- Raising Traps for Events Based on System Log Messages on page 140
- Unsupported Enterprise-Specific SNMP Traps on page 140
- Spoofing Enterprise-Specific SNMP Traps on page 141

## Juniper Networks Enterprise-Specific SNMP Version 1 Traps

The JUNOS software supports enterprise-specific SNMP version 1 traps shown in Table 19 on page 132. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. Traps that do not have corresponding system logging severity levels are marked with an en dash (–).

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 123 and select the corresponding Juniper Networks enterprise-specific MIB. For more information about chassis traps, see “Chassis Traps” on page 375.

Table 19 on page 132 lists the Juniper Networks enterprise-specific supported SNMP version 1 traps.

**Table 19: 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

**Table 19: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps** (continued)

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
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
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	jnxCollFtpAutoSwitchoverToSecondary	1.3.6.1.4.1.2636.4.8	6	8	–	–
Link	jnxCollFtpRequestedSwitchoverToSecondary	1.3.6.1.4.1.2636.4.8	6	9	–	–
Link	jnxCollFtpRequestedSwitchoverToPrimary	1.3.6.1.4.1.2636.4.8	6	10	–	–
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	–	–

**Table 19: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps (continued)**

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
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	jnxPingEgressStdDevThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	5	–	–
Remote operations	jnxPingEgressJitterThresholdExceeded	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	jnxPingIngressStddevThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	8	–	–
Remote operations	jnxPingIngressJitterThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	9	–	–
Routing	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4	6	1	–	–
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	–	–

**Table 19: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps** (continued)

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
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 20 on page 135. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. Traps that do not have corresponding system logging severity levels are marked with an en dash (–).

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 123 and select the corresponding Juniper Networks enterprise-specific MIB. For more information about chassis traps, see “Chassis Traps” on page 375.

**Table 20: 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

**Table 20: Enterprise-Specific Supported SNMP Version 2 Traps** (continued)

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
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
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	–	–

**Table 20: Enterprise-Specific Supported SNMP Version 2 Traps** (continued)

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
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	jnxCollFtpAutoSwitchoverToSecordary	1.3.6.1.4.1.2636.4.8.0.8	–	–
Link	jnxCollFtpRequested SwitchoverToSecondary	1.3.6.1.4.1.2636.4.8.0.9	–	–
Link	jnxCollFtpRequested SwitchoverToPrimary	1.3.6.1.4.1.2636.4.8.0.10	–	–
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	–	–

**Table 20: Enterprise-Specific Supported SNMP Version 2 Traps** (continued)

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
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	jnxPingIngressJitterThresholdExceeded	1.3.6.1.4.1.2636.4.9.0.9	–	–
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	–	–
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	–	–
Routing	jnxAccessAuthServiceUp	1.3.6.1.4.1.2636.3.51.1.0.1	–	–
Routing	jnxAccessAuthServiceDown	1.3.6.1.4.1.2636.3.51.1.0.2	–	–
Routing	jnxAccessAuthServerDisabled	1.3.6.1.4.1.2636.3.51.1.0.3	–	–
Routing	jnxAccessAuthServerEnabled	1.3.6.1.4.1.2636.3.51.1.0.4	–	–
Routing	jnxJsFwAuthFailure	1.3.6.1.4.1.2636.3.39.1.2.1.0.1	–	–
Routing	jnxJsFwAuthServiceUp	1.3.6.1.4.1.2636.3.39.1.2.1.0.2	–	–
Routing	jnxJsFwAuthServiceDown	1.3.6.1.4.1.2636.3.39.1.2.1.0.3	–	–
Routing	nxJsFwAuthCapacityExceeded	1.3.6.1.4.1.2636.3.39.1.2.1.0.4	–	–
Routing	jnxJsNatAddrPoolThresholdStatus	1.3.6.1.4.1.2636.3.39.1.7.1.0.1	–	–



**Table 20: Enterprise-Specific Supported SNMP Version 2 Traps** (continued)

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
Routing	jnxJsScreenAttack	1.3.6.1.4.1.2636.3.39.1.8.1.0.1	Warning	RT_SCREEN_ICMP, RT_SCREEN_IP, RT_SCREEN_SESSION_LIMIT, RT_SCREEN_TCP, RT_SCREEN_UDP
Routing	jnxJsScreenCfgChange	1.3.6.1.4.1.2636.3.39.1.8.1.0.2	–	–
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	–	–

## Juniper Networks Enterprise-Specific LDP Traps

For information on the enterprise-specific LDP traps, see “Interpreting the Enterprise-Specific LDP MIB” on page 631.

## Disabling LDP Traps

You can disable the LDP LSP notifications by including the trap disable statement at the [show protocols ldp log-updown] hierarchy level.

## Juniper Networks Enterprise-Specific Version 2 Traps on EX-Series Ethernet Switches

EX-series Ethernet switches support the following enterprise-specific traps:

- jnxSecAccessdsRateLimitCrossed
- jnxSecIfMacLimitExceeded
- jnxStormEventNotification

For more information about the enterprise-specific traps supported on EX-series, see “Interpreting the Enterprise-Specific Secure Access Port MIB” on page 651.

## Juniper Networks Enterprise-Specific Version 2 Traps on MX960 Platforms

On the MX960 platform, SNMP traps are generated when the MAC address table on a logical interface or on a bridging-domain reaches its maximum number of entries. You can enable or disable the MAC address learning feature and also configure the

maximum number of MAC entries that a logical interface or bridging-domain can store in the MAC address table.

The following traps, defined in the L2ALD MIB, `jnxl2ald.mib`, are generated when the respective MAC limit is reached:

- `jnxl2aldRoutingInstMacLimit`: Generated when the number of MAC addresses for the given routing instance, `jnxl2aldRoutingInst`, exceeds the set limit.
- `jnxl2aldInterfaceMacLimit`: Generated when the number of MAC addresses for the given physical interface exceeds the configured limit.
- `jnxl2aldGlobalMacLimit`: Generated when the number of MAC addresses for the entire system exceeds the configured limit.

## Raising Traps for Events Based on System Log Messages

Event policies can include an action that raises traps for events based on system log messages. This feature enables notification of an SNMP trap-based application when an important system log message occurs. You can convert any system log message (for which there are no corresponding traps) into a trap. This feature is valuable for customers who use network management system traps rather than system log messages to monitor their networks.

For information on converting system log messages into traps, see the *JUNOS Configuration and Diagnostic Automation Guide*. For information on the System Log MIB that provides support for this feature, see “Interpreting the Enterprise-Specific System Log MIB” on page 477.

## Unsupported Enterprise-Specific SNMP Traps

Enterprise-specific SNMP traps that are defined in JUNOS software but are not generated are shown in Table 21 on page 140. For a list of standard traps that are defined in JUNOS software, but are not generated, see “Unsupported Standard SNMP Traps” on page 160.

**Table 21: Unsupported Enterprise-Specific SNMP Traps**

MIB	Trap Name	Description
<code>jnx-bgpmib2.mib</code>	<code>jnxBgpM2Established</code>	Generated when the BGP finite state machine (FSM) enters the Established state.
	<code>jnxBgpM2BackwardTransition</code>	Generated when the BGP finite state machine moves from a higher-numbered state to a lower-numbered state.
<code>jnx-sonetaps.mib</code>	<code>apsEventFEPLF</code>	Generated when the value of an instance of <code>apsStatusFEPLFs</code> increments.

## Spoofing Enterprise-Specific SNMP Traps

---

You can use the `request snmp spoof-trap` operational mode command to mimic SNMP trap behavior. The contents of the traps (the values and instances of the objects carried in the trap) can be specified on the command line or they can be spoofed automatically. This feature is useful if you want to trigger SNMP traps from routers and ensure they are processed correctly within your existing network management infrastructure, but find it difficult to simulate the error conditions that trigger many of the traps on the router. For more information, see the *JUNOS System Basics and Services Command Reference*.



## Chapter 12

# 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 on page 143
- Standard SNMP Version 2 Traps on page 149
- Standard SNMP Traps on EX-Series Ethernet Switches on page 159
- Unsupported Standard SNMP Traps on page 160
- Spoofing Standard SNMP Traps on page 164

### Standard SNMP Version 1 Traps

Table 22 on page 143 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. Traps that do not have corresponding system logging severity levels are marked with an en dash (–) in the table.

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 22: 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	Warning	SNMP_TRAP_LINK_DOWN

**Table 22: Standard Supported SNMP Version 1 Traps** (continued)

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag
Link	linkUp	1.3.6.1.4.1.2636	3	0	Info	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	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2	6	6	–	–
Routing	ospfVirtIfAuthFailure	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	–	–

**Table 22: Standard Supported SNMP Version 1 Traps** (continued)

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag
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	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2	6	13	–	–
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 145
- SNMP Version 1 Ping Traps MIB on page 146
- SNMP Version 1 Traceroute Traps MIB on page 147
- SNMP Version 1 VRRP Traps MIB on page 148

### 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*:

```

coldStartTRAP-TYPE
ENTERPRISEsnmp
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
warmStartTRAP-TYPE
ENTERPRISEsnmp
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
OBJECTS {
    ifIndex

```

```

ifAdminStatus
ifOperStatus
ifName
}
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
OBJECTS {
    ifIndex
    ifAdminStatus
    ifOperStatus
    ifName
}
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
}
STATUSmandatory
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
pingTestFailedTRAP-TYPE
ENTERPRISEpingMIB
VARIABLES {
    pingCtlTargetAddressType, pingCtlTargetAddress,
    pingResultsOperStatus, pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress, pingResultsMinRtt,
    pingResultsMaxRtt, pingResultsAverageRtt,
    pingResultsProbeResponses, pingResultsSentProbes,
    pingResultsRttSumOfSquares, pingResultsLastGoodProbe
}
STATUSmandatory
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
pingTestCompletedTRAP-TYPE
ENTERPRISE pingMIB
VARIABLES {
    pingCtlTargetAddressType, pingCtlTargetAddress,
    pingResultsOperStatus, pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress, pingResultsMinRtt,
    pingResultsMaxRtt, pingResultsAverageRtt,
    pingResultsProbeResponses, pingResultsSentProbes,
    pingResultsRttSumOfSquares, pingResultsLastGoodProbe
}
STATUSmandatory
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
traceRoutePathChangeTRAP-TYPE
```

```

ENTERPRISEtraceRouteMIB
VARIABLES {
    traceRouteCtlTargetAddressType,
    traceRouteCtlTargetAddress,
    traceRouteResultsIpTgtAddrType,
    traceRouteResultsIpTgtAddr
}
STATUSmandatory
DESCRIPTION
"The path to a target has changed."
::= 1
traceRouteTestFailedTRAP-TYPE
ENTERPRISEtraceRouteMIB
VARIABLES {
    traceRouteCtlTargetAddressType,
    traceRouteCtlTargetAddress,
    traceRouteResultsIpTgtAddrType,
    traceRouteResultsIpTgtAddr
}
STATUSmandatory
DESCRIPTION
"Could not determine the path to a target."
::= 2
traceRouteTestCompletedTRAP-TYPE
ENTERPRISEtraceRouteMIB
VARIABLES {
    traceRouteCtlTargetAddressType,
    traceRouteCtlTargetAddress,
    traceRouteResultsIpTgtAddrType,
    traceRouteResultsIpTgtAddr
}
STATUSmandatory
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
vrrpTrapNewMasterTRAP-TYPE
ENTERPRISEvrrpMIB
VARIABLES {
    vrrpOperMasterIpAddr
}
STATUSmandatory
DESCRIPTION
"The newMaster trap indicates that the sending agent has transitioned to
'Master' state."
::= 1
vrrpTrapAuthFailureTRAP-TYPE
ENTERPRISEvrrpMIB
VARIABLES {

```

```

        vrrpTrapPacketSrc
        vrrpTrapAuthErrorType
    }
    STATUSmandatory
    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 23 on page 149 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. Traps that do not have corresponding system logging severity levels are marked with an en dash (–) in the table.

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 23: 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	1.3.6.1.2.1.16.0.1	–	–

**Table 23: Standard Supported SNMP Version 2 Traps** (continued)

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag
RMON alarm	risingAlarm	1.3.6.1.2.1.16.0.2	–	–
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	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2.6	–	–
Routing	ospfVirtIfAuthFailure	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	–	–
Routing	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2.11	–	–
Routing	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2.13	–	–
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_NEWMASER_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 151
- SNMP Version 2 MPLS Traps on page 152
- SNMP Version 2 OSPF Traps MIB on page 153
- SNMP Version 2 Ping Traps MIB on page 157

- SNMP Version 2 Traceroute Traps MIB on page 158
- SNMP Version 2 VRRP Traps MIB on page 159

## 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*:

```

coldStartNOTIFICATION-TYPE
STATUScurrent
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 }
warmStartNOTIFICATION-TYPE
STATUScurrent
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 }
linkDownNOTIFICATION-TYPE
OBJECTS {
    ifIndex
    ifAdminStatus
    ifOperStatus
    ifName
}
STATUScurrent
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 }
linkUpNOTIFICATION-TYPE
OBJECTS {
    ifIndex
    ifAdminStatus
    ifOperStatus
    ifName
}
STATUScurrent
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
STATUScurrent
DESCRIPTION
"An authenticationFailure trap signifies that the SNMPv2 entity, acting in an

```



## 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*:

```
ospflfStateChangeNOTIFICATION-TYPE
OBJECTS {
    ospfRouterId, -- The originator of the trap
    ospflfIpAddress,
    ospfAddressLessIf,
}
STATUS current
DESCRIPTION
    "An ospflfStateChange 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 }

ospfvirtlfStateChange NOTIFICATION-TYPE
OBJECTS {
    ospfRouterId, -- The originator of the trap
    ospfvirtlfAreaId,
    ospfvirtlfNeighbor,
}
STATUS current
DESCRIPTION
    "An ospfvirtlfStateChange 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
}
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 ospflfStateChange."
```

```
::= { ospfTraps 2 }

ospfvirtnbrStateChange NOTIFICATION-TYPE
OBJECTS {
    ospfRouterId, -- The originator of the trap
    ospfvirtnbrArea,
    ospfvirtnbrRtrId,
```

```

ospfVirtNbrState
}
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 ospfVirtIfConfigError 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 ospflfAuthFailure 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
    ospfVirtIfAreId,
    ospfVirtIfNeighbor,
    ospfConfigErrorType, – authTypeMismatch or
    – authFailure
}
STATUScurrent
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 }
ospflfRxBadPacket NOTIFICATION-TYPE
OBJECTS {
    ospfRouterId, – The originator of the trap
    ospflfIpAddress,
    ospfAddressLessIf,
    ospfPacketSrc, – The source IP address
    ospfPacketType
}
STATUScurrent
DESCRIPTION
"An ospflfRxBadPacket 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
    ospfVirtIfAreId,
    ospfVirtIfNeighbor,
    ospfPacketType
}
STATUScurrent
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
    ospflfIpAddress,
    ospfAddressLessIf,
    ospfNbrRtrId, – Destination
    ospfPacketType,

```

```

        ospfLsdbType,
        ospfLsdbLsid,
        ospfLsdbRouterId
    }
    STATUScurrent
    DESCRIPTION
    "An ospfTxRetransmit trap signifies that an OSPF packet has
      been retransmitted
on a nonvirtual 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 10 }
    ospfVirtIfTxRetransmit NOTIFICATION-TYPE
    OBJECTS {
        ospfRouterId, – The originator of the trap
        ospfVirtIfAreald,
        ospfVirtIfNeighbor,
        ospfPacketType,
        ospfLsdbType,
        ospfLsdbLsid,
        ospfLsdbRouterId
    }
    STATUScurrent
    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 }
    ospfMaxAgeLsa NOTIFICATION-TYPE
    OBJECTS {
        ospfRouterId, – The originator of the trap
        ospfLsdbAreald, – 0.0.0.0 for AS Externals
        ospfLsdbType,
        ospfLsdbLsid,
        ospfLsdbRouterId
    }
    STATUScurrent
    DESCRIPTION
    "An ospfMaxAgeLsa trap signifies that one of the LSAs
in the router's link-state
database has aged to MaxAge."
 ::= { ospfTraps 13 }
    }
}
}
}
}
}
}
}
}
}
}

```

```

    }
  }
}

```

## 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*:

```

pingProbeFailedNOTIFICATION-TYPE
OBJECTS {
    pingCtlTargetAddressType,
    pingCtlTargetAddress,
    pingResultsOperStatus,
    pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress,
    pingResultsMinRtt,
    pingResultsMaxRtt,
    pingResultsAverageRtt,
    pingResultsProbeResponses,
    pingResultsSentProbes,
    pingResultsRttSumOfSquares,
    pingResultsLastGoodProbe
}
STATUScurrent
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 }
pingTestFailedNOTIFICATION-TYPE
OBJECTS {
    pingCtlTargetAddressType,
    pingCtlTargetAddress,
    pingResultsOperStatus,
    pingResultsIpTargetAddressType,
    pingResultsIpTargetAddress,
    pingResultsMinRtt,
    pingResultsMaxRtt,
    pingResultsAverageRtt,
    pingResultsProbeResponses,
    pingResultsSentProbes,
    pingResultsRttSumOfSquares,
    pingResultsLastGoodProbe
}
STATUScurrent
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 }
 pingTestCompletedNOTIFICATION-TYPE
 OBJECTS {
   pingCtlTargetAddressType,
   pingCtlTargetAddress,
   pingResultsOperStatus,
   pingResultsIpTargetAddressType,
   pingResultsIpTargetAddress,
   pingResultsMinRtt,
   pingResultsMaxRtt,
   pingResultsAverageRtt,
   pingResultsProbeResponses,
   pingResultsSentProbes,
   pingResultsRttSumOfSquares,
   pingResultsLastGoodProbe
 }
 STATUScurrent
 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*:

```

traceRoutePathChangeNOTIFICATION-TYPE
 OBJECTS {
   traceRouteCtlTargetAddressType,
   traceRouteCtlTargetAddress,
   traceRouteResultsIpTgtAddrType,
   traceRouteResultsIpTgtAddr
 }
 STATUScurrent
 DESCRIPTION
 "The path to a target has changed."
 ::= { traceRouteNotifications 1 }
 traceRouteTestFailedNOTIFICATION-TYPE
 OBJECTS {
   traceRouteCtlTargetAddressType,
   traceRouteCtlTargetAddress,
   traceRouteResultsIpTgtAddrType,
   traceRouteResultsIpTgtAddr
 }
 STATUScurrent
 DESCRIPTION
 "Could not determine the path to a target."
 ::= { traceRouteNotifications 2 }
 traceRouteTestCompletedNOTIFICATION-TYPE
 OBJECTS {

```

```

        traceRouteCtlTargetAddressType,
        traceRouteCtlTargetAddress,
        traceRouteResultsIpTgtAddrType,
        traceRouteResultsIpTgtAddr
    }
    STATUScurrent
    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
vrrpTrapPacketSrcOBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESSaccessible-for-notify
STATUScurrent
DESCRIPTION
"The IP address of an inbound VRRP packet. Used by
vrrpTrapAuthFailure trap."
::= { vrrpOperations 5 }
vrrpTrapAuthErrorTypeOBJECT-TYPE
SYNTAXINTEGER {
    invalidAuthType (1),
    authTypeMismatch (2),
    authFailure (3)
}
MAX-ACCESSaccessible-for-notify
STATUScurrent
DESCRIPTION
"Potential types of configuration conflicts. Used by vrrpAuthFailure trap."
}

```

## Standard SNMP Traps on EX-Series Ethernet Switches

Apart from the standard SNMP traps listed in the preceding sections, JUNOS software also supports the following standard traps on the EX-series Ethernet switches:

- **ptopoConfigChange**—Generated when the value of **ptopoLastChangeTime** changes. Enables a network management system to trigger physical topology table maintenance polls. (RFC 2622, *PTOPO MIB*)
- **pethPsePortOnOffNotification**—Generated when the power supply status of a PSE port changes. Indicates whether the PSE port is supplying power to the PD port or not. (RFC 3621 *Power Ethernet MIB*)

- **pethMainPowerUsageOnNotification**—Generated when the PSE threshold indicator is on. (RFC 3621 *Power Ethernet MIB*)
- **pethMainPowerUsageOffNotification**—Generated when the PSE threshold indicator is off. (RFC 3621 *Power Ethernet MIB*)

## Unsupported Standard SNMP Traps

---

Standard SNMP traps that are defined in JUNOS software but are not generated are shown in Table 24 on page 161. For a list of enterprise-specific traps that are defined in JUNOS software, but are not generated, see “Unsupported Enterprise-Specific SNMP Traps” on page 140.

**Table 24: Unsupported Standard SNMP Traps**

MIB	Trap Name	Description
isismib.mib	isisDatabaseOverload	Generated when the system enters or leaves the overload state.
	isisManualAddressDrops	Generated when one of the manual <code>areaAddresses</code> assigned to the system is ignored when computing routes.
	isisCorruptedLSPDetected	Generated when an LSP stored in memory becomes corrupted.
	isisAttemptToExceedMaxSequence	Generated when the sequence number on a generated LSP wraps the 32-bit sequence counter and the number is purged.
	isisIDLenMismatch	Generated when a PDU is received with a different value for the system ID length. This trap includes an index to identify the circuit where the PDU was received and the PDU header.
	isisMaxAreaAddressesMismatch	Generated when a PDU with a different value for the maximum area addresses is received.
	isisOwnLSPPurge	Generated when a PDU is received with a system ID and zero age. This notification includes the circuit index if available.
	isisSequenceNumberSkip	Generated when an LSP is received with a system ID and different contents, indicating the LSP may require a higher sequence number.
	isisAuthenticationTypeFailure	Generated when a PDU with the wrong authentication type field is received.
	isisAuthenticationFailure	Generated when a PDU with an incorrect authentication information field is received.
	isisVersionSkew	Generated when a hello PDU from an IS running a different version of the protocol is received.
	isisAreaMismatch	Generated when a hello PDU from an IS which does not share any area address is received.
	isisRejectedAdjacency	Generated when a hello PDU from an IS is received, but no adjacency is established because of a lack of resources.
	isisLSPTooLargeToPropagate	Generated when an LSP which is larger than the <code>dataLinkBlockSize</code> for a circuit is attempted, but not propagated.
	isisOriginatingLSPBufferSizeMismatch	Generated when a Level 1 LSP or Level 2 LSP is received that is larger than the local value for originating <code>L1LSPBufferSize</code> or originating <code>L2LSPBufferSize</code> , respectively, or when a Level 1 LSP or Level 2 LSP is received containing the originating <code>LSPBufferSize</code> option and the value in the PDU option field does not match the local value for originating <code>L1LSPBufferSize</code> or originating <code>L2LSPBufferSize</code> , respectively.
	isisProtocolsSupportedMismatch	Generated when a non-pseudonode, segment 0 LSP is received that has no matching protocols.

**Table 24: Unsupported Standard SNMP Traps** (continued)

MIB	Trap Name	Description
l3vpn-mib.mib	mplsVrflfUp	Generated when the <code>ifOperStatus</code> of an interface associated with a VRF changes to the <code>up(1)</code> state, or when an interface with <code>ifOperStatus = up(1)</code> is associated with a VRF.
	mplsVrflfDown	Generated when the <code>ifOperStatus</code> of an interface associated with a VRF changes to the <code>down(1)</code> state, or when an interface with <code>ifOperStatus = up(1)</code> state is disassociated from a VRF.
	mplsNumVrfRouteMidThreshExceeded	Generated when the number of routes contained by the specified VRF exceeds the value indicated by <code>mplsVrfMidRouteThreshold</code> .
	mplsNumVrfRouteMaxThreshExceeded	Generated when the number of routes contained by the specified VRF reaches or attempts to exceed the maximum allowed value as indicated by <code>mplsVrfMaxRouteThreshold</code> .
	mplsNumVrfSecIllegalLblThreshExcd	Generated when the number of illegal label violations on a VRF as indicated by <code>mplsVpnVrfSecIllegalLblVtns</code> has exceeded <code>mplsVpnVrfSecIllegalLblRcvThresh</code> .
ldp-mib.mib	mplsLdpInitSesThresholdExceeded	Generated when the value of <code>mplsLdpEntityInitSesThreshold</code> is not zero and the number of session initialization messages exceeds the value of <code>mplsLdpEntityInitSesThreshold</code> .
	mplsLdpPathVectorLimitMismatch	Generated when the <code>mplsLdpEntityPathVectorLimit</code> does not match the value of the <code>mplsLdpPeerPathVectorLimit</code> for a specific entity.
	mplsLdpSessionUp	Generated when the value of <code>mplsLdpSesState</code> enters the <code>operational(5)</code> state.
	mplsLdpSessionDown	Generated when the value of <code>mplsLdpSesState</code> leaves the <code>operational(5)</code> state.
msdp-mib.mib	msdpEstablished	Generated when the MSDP FSM enters the <b>Established</b> state.
	msdpBackwardTransition	Generated when the MSDP FSM moves from a higher numbered state to a lower numbered state.



**Table 24: Unsupported Standard SNMP Traps** (continued)

MIB	Trap Name	Description
ospf2trap.mib	ospfVirtualIfConfigError	Generated when a packet is received on a virtual interface from a router whose configuration parameters conflict with the receiving router's configuration parameters.
	ospfVirtualIfAuthFailure	Generated when a packet is received on a virtual interface from a router whose authentication key or authentication type conflicts with the receiving router's authentication key or authentication type.
	ospfVirtualIfRxBadPacket	Generated when an OSPF packet is received on a virtual interface and cannot be parsed.
	ospfOriginateLsa	Generated when a new LSA is originated by the router because of a topology change.
	ospfLsdbOverflow	Generated when the number of LSAs in the router's link-state database exceeds the value of <code>ospfExtLsdbLimit</code> .
	ospfLsdbApproachingOverflow	Generated when the number of LSAs in the router's link-state database exceeds 90 % of the value of <code>ospfExtLsdbLimit</code> .
rfc1747.mib	sdlcPortStatusChange	Generated when the state of an SDLC port transitions to active or inactive.
	sdlcLSStatusChange	Generated when the state of an SDLC link station transitions to contacted or disconnected.
rfc2115a.mib	frDLCIStatusChange	Generated when a virtual circuit changes state (has been created or invalidated, or has toggled between the active and inactive states).
rfc2662.mib	adslAtucRateChangeTrap	Generated when the ATUCs transmit rate has changed (RADSL mode only).
	adslAtucPerfLofsThreshTrap	Generated when the loss of framing 15-minute interval threshold is reached.
	adslAtucInitFailureTrap	Generated when ATUC initialization fails.
	adslAturPerfLprsThreshTrap	Generated when the loss of power 15-minute interval threshold is reached.
	adslAturRateChangeTrap	Generated when the ATURs transmit rate changes (RADSL mode only).
rfc3020.mib	mfrMibTrapBundleLinkMismatch	Generated when a bundle link mismatch is detected.
rfc3813.mib	mplsXCUp	Generated when <code>mplsXCOperStatus</code> for one or more contiguous entries in <code>mplsXCTable</code> enters the <code>up(1)</code> state from some other state.
	mplsXCDown	Generated when <code>mplsXCOperStatus</code> for one or more contiguous entries in <code>mplsXCTable</code> enters the <code>down(2)</code> state from some other state.

## Spoofing Standard SNMP Traps

---

You can use the `request snmp spoof-trap` operational mode command to mimic SNMP trap behavior. The contents of the traps (the values and instances of the objects carried in the trap) can be specified on the command line or they can be spoofed automatically. This feature is useful if you want to trigger SNMP traps from routers and ensure they are processed correctly within your existing network management infrastructure, but find it difficult to simulate the error conditions that trigger many of the traps on the router. For more information, see the *JUNOS System Basics and Services Command Reference*.

## Chapter 13

# 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

---

<b>Syntax</b>	agent-address outgoing-interface;
<b>Hierarchy Level</b>	[edit snmp trap-options]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Set the agent address of all SNMPv1 traps generated by this router. Currently, the only option is <b>outgoing-interface</b> , which sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.
<b>Options</b>	<b>outgoing-interface</b> —Value of agent address of all SNMPv1 traps generated by this router. The <b>outgoing-interface</b> option sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap. <b>Default:</b> disabled (The agent address is not specified in SNMPv1 traps.)
<b>Usage Guidelines</b>	See “Configuring the Agent Address for SNMP Traps” on page 41.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## authorization

---

<b>Syntax</b>	<code>authorization <i>authorization</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp community <i>community-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Set the access authorization for SNMP <code>Get</code> , <code>GetBulk</code> , <code>GetNext</code> , and <code>Set</code> requests.
<b>Options</b>	<p><i>authorization</i>—Access authorization level:</p> <ul style="list-style-type: none"> <li>■ <code>read-only</code>—Enable <code>Get</code>, <code>GetNext</code>, and <code>GetBulk</code> requests.</li> <li>■ <code>read-write</code>—Enable all requests, including <code>Set</code> requests. You must configure a view to enable <code>Set</code> requests.</li> </ul> <p><b>Default:</b> <code>read-only</code></p>
<b>Usage Guidelines</b>	See “Configuring the SNMP Community String” on page 36.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

## categories

---

<b>Syntax</b>	<pre>categories {   <i>category</i>; }</pre>
<b>Hierarchy Level</b>	<code>[edit snmp trap-group <i>group-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define the types of traps that will be sent to the targets of the named trap group.
<b>Default</b>	If you omit the <code>categories</code> statement, all trap types are included in trap notifications.
<b>Options</b>	<p><i>category</i>—Name of a trap type.</p> <p><b>Values:</b> <code>authentication</code>, <code>chassis</code>, <code>configuration</code>, <code>link</code>, <code>remote-operations</code>, <code>rmon-alarm</code>, <code>routing</code>, <code>sonet-alarms</code>, <code>startup</code>, <code>vrp-events</code></p>
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

## client-list

---

<b>Syntax</b>	<code>client-list <i>client-list-name</i> {     <i>ip-addresses</i>; }</code>
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.5.
<b>Description</b>	Define a list of SNMP clients.
<b>Options</b>	<i>client-list-name</i> —Name of the client list.  <i>ip-addresses</i> —IP addresses of the SNMP clients to be added to the client list,
<b>Usage Guidelines</b>	See “Adding a Group of Clients to an SNMP Community” on page 37.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## client-list-name

---

<b>Syntax</b>	<code>client-list-name <i>client-list-name</i>;</code>
<b>Hierarchy Level</b>	[edit snmp community <i>community-name</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.5.
<b>Description</b>	Add a client list or prefix list to an SNMP community.
<b>Options</b>	<i>client-list-name</i> —Name of the client list or prefix list.
<b>Usage Guidelines</b>	See “Adding a Group of Clients to an SNMP Community” on page 37.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## clients

---

<b>Syntax</b>	clients { address <restrict>; }
<b>Hierarchy Level</b>	[edit snmp community <i>community-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the IPv4 or IPv6 addresses of the SNMP client hosts that are authorized to use this community.
<b>Default</b>	If you omit the <b>clients</b> statement, all SNMP clients using this community string are authorized to access the router.
<b>Options</b>	<p><b>address</b>—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 <b>address</b> options.</p> <p><b>restrict</b>—(Optional) Do not allow the specified SNMP client to access the router.  <b>Default:</b> The client is granted access.</p>
<b>Usage Guidelines</b>	See “Configuring the SNMP Community String” on page 36.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## commit-delay

---

<b>Syntax</b>	commit-delay <i>seconds</i> ;
<b>Hierarchy Level</b>	[edit snmp nonvolatile]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the timer for the SNMP Set reply and start of the commit.
<b>Default</b>	5 seconds
<b>Usage Guidelines</b>	See “Configuring the Commit Delay Timer” on page 35.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## community

---

**Syntax**    `community community-name {  
                   authorization authorization;  
                   client-list-name client-list-name;  
                   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 36.

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

## contact

---

<b>Syntax</b>	contact <i>contact</i> ;
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define the value of the MIB II <b>sysContact</b> object, which is the contact person for the managed system.
<b>Options</b>	<i>contact</i> —Name of contact person. If the name includes spaces, enclose it in quotation marks (" ").
<b>Usage Guidelines</b>	See “Configuring the System Contact on a JUNOS Device” on page 34.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## description

---

<b>Syntax</b>	description <i>description</i> ;
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define the value of the MIB II <b>sysDescription</b> object, which is the description of the system being managed.
<b>Options</b>	<i>description</i> —System description. If the name includes spaces, enclose it in quotation marks (" ").
<b>Usage Guidelines</b>	See “Configuring the System Description on a JUNOS Device” on page 34.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.



## destination-port

---

<b>Syntax</b>	destination-port <i>port-number</i> ;
<b>Hierarchy Level</b>	[edit snmp trap-group]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Assign a trap port number other than the default.
<b>Default</b>	If you omit this statement, the default port is 162.
<b>Options</b>	<i>port-number</i> —SNMP trap port number.
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## engine-id

---

**See** engine-id

## filter-duplicates

---

<b>Syntax</b>	filter-duplicates;
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Filter duplicate Get, GetNext, or GetBulk SNMP requests.
<b>Usage Guidelines</b>	See “Filtering Duplicate SNMP Requests” on page 35.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## interface

---

<b>Syntax</b>	interface [ <i>interface-names</i> ];
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the interfaces on which SNMP requests can be accepted.
<b>Default</b>	If you omit this statement, SNMP requests entering the router through any interface will be accepted.
<b>Options</b>	<i>interface-names</i> —Names of one or more logical interfaces.
<b>Usage Guidelines</b>	See “Configuring the Interfaces on Which SNMP Requests Can Be Accepted” on page 44.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## location

---

<b>Syntax</b>	location <i>location</i> ;
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define the value of the MIB II <b>sysLocation</b> object, which is the physical location of the managed system.
<b>Options</b>	<i>location</i> —Location of the local system. You must enclose the name within quotation marks (" ").
<b>Usage Guidelines</b>	See “Configuring the System Location for a JUNOS Device” on page 34.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## logical-system

---

**Syntax**    `logical-system logical-system-name {  
                  routing-instance routing-instance-name;  
                  }`

**Hierarchy Level**    `[edit snmp community community-name],  
                          [edit snmp trap-group],  
                          [edit snmp trap-options]`

**Release Information**    Statement introduced in JUNOS Release 9.3



**NOTE:** The `logical-system` statement replaces the `logical-router` statement, and is backward compatible with the JUNOS software versions not lower than Release 8.3.

---

**Description**    Specify a logical system name for SNMP v1 and v2c clients.

**Options**    `logical-system-name`—Name of the logical system.  
  
`routing-instance routing-instance-name`—Statement to specify a routing instance associated with the logical system.

**Usage Guidelines**    See “Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community” on page 108.

**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 35.

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

## nonvolatile

---

<b>Syntax</b>	nonvolatile { commit-delay <i>seconds</i> ; }
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure options for SNMP Set requests.  The statement is explained separately in this chapter.
<b>Usage Guidelines</b>	See “Configuring the Commit Delay Timer” on page 35.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## oid

---

<b>Syntax</b>	oid <i>object-identifier</i> (include   exclude);
<b>Hierarchy Level</b>	[edit snmp view <i>view-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify an object identifier (OID) used to represent a subtree of MIB objects.
<b>Options</b>	<p><i>object-identifier</i>—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.</p> <p>include—Include the subtree of MIB objects represented by the specified OID.</p> <p>exclude—Exclude the subtree of MIB objects represented by the specified OID.</p>
<b>Usage Guidelines</b>	See “Configuring MIB Views” on page 45.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## routing-instance

---

<b>Syntax</b>	<code>routing-instance routing-instance-name;</code>
<b>Hierarchy Level</b>	[edit snmp community <i>community-name</i> ], [edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i> ], [edit snmp trap-group <i>group</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.3. Added to [edit snmp community <i>community-name</i> ] hierarchy level in JUNOS Release 8.4 Added to [edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i> ] in JUNOS Release 9.1
<b>Description</b>	Specify a routing instance for SNMPv1 and SNMPv2 trap targets. All targets configured in the trap group use this routing instance.
<b>Options</b>	<i>routing-instance-name</i> —Name of the routing instance.  If the routing instance is defined within a logical system, include the <i>logical-system logical-system-name</i> statement at the [edit snmp community <i>community-name</i> ] hierarchy level and then, specify the <i>routing-instance</i> statement under the [edit snmp community <i>community-name</i> logical-system <i>logical system-name</i> ] hierarchy level.
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41 and “Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community” on page 108.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## snmp

---

<b>Syntax</b>	<code>snmp { ... } }</code>
<b>Hierarchy Level</b>	[edit]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure SNMP.
<b>Usage Guidelines</b>	See “Configuring SNMP” on page 31.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## source-address

---

<b>Syntax</b>	source-address <i>address</i> ;
<b>Hierarchy Level</b>	[edit snmp trap-options]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	<p><i>address</i>—Source address of SNMP traps. You can configure the source address of trap packets two ways: <b>lo0</b> or a valid IPv4 address configured on one of the router interfaces. The value <b>lo0</b> indicates that the source address of all SNMP trap packets will be set to the lowest loopback address configured at interface <b>lo0</b>.</p> <p><b>Default:</b> disabled (The source address is the address of the outgoing interface.)</p>
<b>Usage Guidelines</b>	See “Configuring the Source Address for SNMP Traps” on page 40.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## targets

---

<b>Syntax</b>	targets { <i>address</i> ; }
<b>Hierarchy Level</b>	[edit snmp trap-group <i>group-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure one or more systems to receive SNMP traps.
<b>Options</b>	<i>address</i> —IPv4 or IPv6 address of the system to receive traps. You must specify an address, not a hostname.
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## traceoptions

---

<b>Syntax</b>	<pre> traceoptions {     file <i>filename</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;&lt;match         <i>regex</i>&gt;;     flag <i>flag</i>; } </pre>
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	<p>Statement introduced before JUNOS Release 7.4.</p> <p>file <i>filename</i> option added in JUNOS Release 8.1.</p> <p>world-readable   no-world-readable option added in JUNOS Release 8.1.</p> <p>match <i>regex</i> option added in JUNOS Release 8.1.</p>
<b>Description</b>	<p>The output of the tracing operations is placed into log files in the <code>/var/log</code> directory. Each log file is named after the SNMP agent that generates it. Currently, the following logs are created in the <code>/var/log</code> directory when the <b>traceoptions</b> statement is used:</p> <ul style="list-style-type: none"> <li>■ chassisd</li> <li>■ craftd</li> <li>■ ilmid</li> <li>■ mib2d</li> <li>■ rmopd</li> <li>■ serviced</li> <li>■ snmpd</li> </ul>
<b>Options</b>	<p>file <i>filename</i>—By default, the name of the log file that records trace output is the name of the process being traced (for example, <b>mib2d</b> or <b>snmpd</b>). Use this option to specify another name.</p> <p>world-readable   no-world-readable—(Optional) By default, log files can be accessed only by the user who configures the tracing operation. The <b>world-readable</b> option enables any user to read the file. To explicitly set the default behavior, use the <b>no-world-readable</b> option.</p> <p>files <i>number</i>—(Optional) Maximum number of trace files per SNMP subagent. When a trace file (for example, <b>snmpd</b>) reaches its maximum size, it is archived by being renamed to <b>snmpd.0</b>. The previous <b>snmpd.1</b> is renamed to <b>snmpd.2</b>, and so on. The oldest archived file is deleted.</p> <p><b>Range:</b> 2 through 1000 files</p> <p><b>Default:</b> 10 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple <b>flag</b> statements:</p> <ul style="list-style-type: none"> <li>■ all—Log all SNMP events.</li> </ul>

- **configuration**—Log reading of configuration at the `[edit snmp]` hierarchy level.
- **database**—Log events involving storage and retrieval in the events database.
- **events**—Log important events.
- **general**—Log general events.
- **interface-stats**—Log physical and logical interface statistics.
- **nonvolatile-sets**—Log nonvolatile SNMP set request handling.
- **pdu**—Log SNMP request and response packets.
- **policy**—Log policy processing.
- **protocol-timeouts**—Log SNMP response timeouts.
- **routing-socket**—Log routing socket calls.
- **server**—Log communication with processes that are generating events.
- **subagent**—Log subagent restarts.
- **timer-events**—Log internally generated events.
- **varbind-error**—Log variable binding errors.

**match *regex***—(Optional) Refine the output to include lines that contain the regular expression.

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

**Range:** 10 KB through 1 GB

**Default:** 1000 KB

**Usage Guidelines** See “Tracing SNMP Activity on a JUNOS Device” on page 46.

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



## trap-group

---

<b>Syntax</b>	<pre> trap-group <i>group-name</i> {     categories {         <i>category</i>;     }     destination-port <i>port-number</i>;     routing-instance <i>instance</i>;     targets {         <i>address</i>;     }     version (all   v1   v2); } </pre>
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	<p><i>group-name</i>—Name of the trap group. If the name includes spaces, enclose it in quotation marks (" ").</p> <p>The remaining statements are explained separately.</p>
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## trap-options

---

<b>Syntax</b>	trap-options { agent-address outgoing-interface; source-address <i>address</i> ; }
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	The remaining statements are explained separately. <b>Default:</b> disabled
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## version

---

<b>Syntax</b>	version (all   v1   v2);
<b>Hierarchy Level</b>	[edit snmp trap-group <i>group-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the version number of SNMP traps.
<b>Options</b>	all—Send an SNMPv1 and SNMPv2 trap for every trap condition.  v1—Send SNMPv1 traps only.  v2—Send SNMPv2 traps only. <b>Default:</b> all
<b>Usage Guidelines</b>	See “Configuring SNMP Trap Groups” on page 41.
<b>Required Privilege Level</b>	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 181
- view (Configuring MIB View) on page 182

### ***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 36.

**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 45.

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

**Related Topics**    community

## Chapter 14

# Summary of SNMPv3 Configuration Statements

The following sections explain each of the SNMPv3 configuration statements. The statements are organized alphabetically.

### address

---

<b>Syntax</b>	<code>address <i>address</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the SNMP target address.
<b>Options</b>	<i>address</i> —IPv4 address of the system to receive traps or informs. You must specify an address, not a hostname.
<b>Usage Guidelines</b>	See “Configuring the Address” on page 71.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration


## address-mask

---

<b>Syntax</b>	<code>address-mask <i>address-mask</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Verify the source addresses for a group of target addresses.
<b>Options</b>	<i>address-mask</i> combined with the address defines a range of addresses.
<b>Usage Guidelines</b>	See “Configuring the Address Mask” on page 71.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration


## authentication-md5

---

<b>Syntax</b>	<code>authentication-md5 {     authentication-password <i>authentication-password</i>; }</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 usm local-engine user <i>username</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure MD5 as the authentication type for the SNMPv3 user.
<b>Options</b>	<p><i>authentication-password</i>—Password that generates the key used for authentication.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<hr/> <div style="display: flex; align-items: center;">  <div> <b>NOTE:</b> You can only configure one authentication type for each SNMPv3 user. </div> </div> <hr/>	
<b>Usage Guidelines</b>	See “Configuring MD5 Authentication” on page 58.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## authentication-none

---

<b>Syntax</b>	authentication-none;
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure no authentication for the SNMPv3 user.
<hr/>	
	<b>NOTE:</b> You can only configure one authentication type for each SNMPv3 user.
<hr/>	
<b>Usage Guidelines</b>	See “Configuring No Authentication” on page 59.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## authentication-password

---

<b>Syntax</b>	authentication-password <i>authentication-password</i> ;
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> authentication-md5], [edit snmp v3 usm local-engine user <i>username</i> authentication-sha]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure password for authentication.
<b>Options</b>	<p><i>authentication-password</i>—Password used to generate the key used for authentication.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<b>Usage Guidelines</b>	See “Configuring MD5 Authentication” on page 58 and “Configuring SHA Authentication” on page 58.
<b>Required Privilege Level</b>	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 eight characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

**Usage Guidelines**     See “Configuring SHA Authentication” on page 58.

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



## community-name

---

<b>Syntax</b>	community-name <i>community-name</i> ;
<b>Hierarchy Level</b>	[edit snmp v3 snmp-community <i>community-index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	<i>community-name</i> —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.

---

<b>Usage Guidelines</b>	See “Configuring the SNMPv3 Community” on page 80.
<b>Required Privilege Level</b>	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*—Explicit setting for the engine ID suffix.

              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 56.

**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 \(Configuring\)](#) on page 189
- [group \(Defining Access Privileges for an SNMPv3 Group\)](#) on page 189

### **group (Configuring)**

<b>Syntax</b>	<code>group group-name;</code>
<b>Hierarchy Level</b>	[edit snmp v3 vacm access]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Assign the security name to a group.
<b>Options</b>	<i>group-name</i> —SNMPv3 group name created for the SNMPv3 group.
<b>Usage Guidelines</b>	See “Configuring the Group” on page 63.
<b>Required Privilege Level</b>	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)**

<b>Syntax</b>	<code>group group-name;</code>
<b>Hierarchy Level</b>	[edit snmp v3 vacm security-to-group security-model (usm   v1   v2c) security-name <i>security-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define access privileges granted to a group.
<b>Options</b>	<i>group-name</i> —Identifies a collection of SNMP security names that belong to the same access policy SNMP.
<b>Usage Guidelines</b>	See “Configuring the Group” on page 67.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## inform-retry-count

---

<b>Syntax</b>	inform-retry-count <i>number</i> ;
<b>Hierarchy Level</b>	[edit snmp v3 target-address <i>target-address-name</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 7.4.
<b>Description</b>	Configure the retry count for SNMP informs.
<b>Options</b>	<i>number</i> —Maximum number of times the inform is transmitted if no acknowledgment is received. If no acknowledgment is received after the inform is transmitted the maximum number of times, the inform message is discarded. <b>Default:</b> 3 times
<b>Usage Guidelines</b>	See “Configuring SNMP Informs” on page 76.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	inform-timeout

## inform-timeout

---

<b>Syntax</b>	inform-timeout <i>seconds</i> ;
<b>Hierarchy Level</b>	[edit snmp v3 target-address <i>target-address-name</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 7.4.
<b>Description</b>	Configure the timeout period (in seconds) for SNMP informs.
<b>Options</b>	<i>seconds</i> —Number of seconds to wait for an inform acknowledgment. If no acknowledgment is received within the timeout period, the inform is retransmitted. <b>Default:</b> 15
<b>Usage Guidelines</b>	See “Configuring SNMP Informs” on page 76.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	inform-retry-count

## 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 57.

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

## message-processing-model

---

<b>Syntax</b>	message-processing-model (v1   v2c   v3);
<b>Hierarchy Level</b>	[edit snmp v3 target-parameters <i>target-parameter-name</i> parameters]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the message processing model to be used when generating SNMP notifications.
<b>Options</b>	v1—SNMPv1 message process model. v2c—SNMPv2c message process model. v3—SNMPv3 message process model.
<b>Usage Guidelines</b>	See “Configuring the Message Processing Model” on page 75.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## notify

---

<b>Syntax</b>	<pre>notify name {     tag tag-name;     type (trap   inform); }</pre>
<b>Hierarchy Level</b>	[edit snmp v3]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4. type inform option added in JUNOS Release 7.4.
<b>Description</b>	Select management targets for notifications as well as the type of notifications. Notifications can be either traps or informs.
<b>Options</b>	<i>name</i> —Name assigned to the notification. <i>tag-name</i> —Notifications are sent to all targets configured with this tag. <i>type</i> —Notification type is <b>trap</b> or <b>inform</b> . Traps are unconfirmed notifications. Informs are confirmed notifications.
<b>Usage Guidelines</b>	See “Configuring the SNMPv3 Trap Notification” on page 69 and “Configuring the Inform Notification Type and Target Address” on page 78.
<b>Required Privilege Level</b>	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 \(Applying to Management Target\)](#) on page 193
- [notify-filter \(Configuring\)](#) on page 193

### ***notify-filter (Applying to Management Target)***

<b>Syntax</b>	<code>notify-filter <i>profile-name</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 target-parameters <i>target-parameters-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the notify filter to be used by a specific set of target parameters.
<b>Options</b>	<i>profile-name</i> —Name of the notify filter to apply to notifications.
<b>Usage Guidelines</b>	See “Applying the Trap Notification Filter” on page 74.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

### ***notify-filter (Configuring)***

<b>Syntax</b>	<code>notify-filter <i>profile-name</i> {     oid <i>oid</i> (include   exclude); }</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	<i>profile-name</i> —Name assigned to the notify filter.  The remaining statement is explained separately.
<b>Usage Guidelines</b>	See “Configuring the Trap Notification Filter” on page 70.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	oid

## notify-view

---

<b>Syntax</b>	<code>notify-view <i>view-name</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 vacm access group <i>group-name</i> default-context-prefix security-model (any   usm   v1   v2c) security-level (authentication   none   privacy)]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Associate the view with a community or a group name (SNMPv3).
<b>Options</b>	<i>view-name</i> —Name of the view to which the SNMP user group has access.
<b>Usage Guidelines</b>	See “Configuring the Notify View” on page 65.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	See Configuring MIB Views on page 45.

## oid

---

<b>Syntax</b>	<code>oid <i>oid</i> (include   exclude);</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 notify-filter <i>profile-name</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify an object identifier (OID) used to represent a subtree of MIB objects.
<b>Options</b>	<p><i>oid</i>—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.</p> <p><i>include</i>—Include the subtree of MIB objects represented by the specified OID.</p> <p><i>exclude</i>—Exclude the subtree of MIB objects represented by the specified OID.</p>
<b>Usage Guidelines</b>	See “Configuring the Trap Notification Filter” on page 70.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.



## parameters

---

<b>Syntax</b>	<pre>parameters {   message-processing-model (v1   v2c   v3);   security-model (usm   v1   v2c);   security-level (none   authentication   privacy);   security-name <i>security-name</i>; }</pre>
<b>Hierarchy Level</b>	[edit snmp v3 target-parameters <i>target-parameters-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	<p>Configure a set of target parameters.</p> <p>The remaining statements are explained separately.</p>
<b>Usage Guidelines</b>	See “Defining and Configuring the Trap Target Parameters” on page 74.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## port

---

<b>Syntax</b>	port <i>port-number</i> ;
<b>Hierarchy Level</b>	[edit snmp v3 target-address <i>target-address-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure a UDP port number for an SNMP target.
<b>Default</b>	If you omit this statement, the default port is 162.
<b>Options</b>	<i>port-number</i> —Port number for the SNMP target.
<b>Usage Guidelines</b>	See “Configuring the Port” on page 72.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## privacy-3des

---

<b>Syntax</b>	<pre>privacy-3des {     privacy-password <i>privacy-password</i>; }</pre>
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the triple Data Encryption Standard (3DES) for the SNMPv3 user.
<b>Options</b>	<p><i>privacy-password</i>—The password used to generate the key used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<b>Usage Guidelines</b>	See “Configuring the Encryption Type” on page 59.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

**privacy-aes128**

---

<b>Syntax</b>	<pre>privacy-aes128 {     privacy-password <i>privacy-password</i>; }</pre>
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the Advanced Encryption Standard encryption algorithm (CFB128-AES-128 Privacy Protocol) for the SNMPv3 user.
<b>Options</b>	<p><i>privacy-password</i>—The password used to generate the key used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<b>Usage Guidelines</b>	See “Configuring the Encryption Type” on page 59.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## privacy-des

---

<b>Syntax</b>	privacy-des { privacy-password <i>privacy-password</i> ; }
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure Data Encryption Standard (DES) for the SNMPv3 user.
<b>Options</b>	<p><i>privacy-password</i>—The password used to generate the key used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<b>Usage Guidelines</b>	See “Configuring the Encryption Type” on page 59.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## privacy-none

---

<b>Syntax</b>	privacy-none;
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure no encryption for the SNMPv3 user.
<b>Usage Guidelines</b>	See “Configuring the Encryption Type” on page 59.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## privacy-password

---

<b>Syntax</b>	<code>privacy-password <i>privacy-password</i>;</code>
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine user <i>username</i> privacy-3des], [edit snmp v3 usm local-engine user <i>username</i> privacy-aes128], [edit snmp v3 usm local-engine user <i>username</i> privacy-des]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure a privacy password for the SNMPv3 user.
<b>Options</b>	<p><i>privacy-password</i>—The password used to generate the key used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a routing platform:</p> <ul style="list-style-type: none"> <li>■ The password must be at least eight characters long.</li> <li>■ You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.</li> </ul>
<b>Usage Guidelines</b>	See “Configuring the Encryption Type” on page 59.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## read-view

---

<b>Syntax</b>	<code>read-view <i>view-name</i>;</code>
<b>Hierarchy Level</b>	[edit snmp v3 vacm access group <i>group-name</i> default-context-prefix security-model (any   usm   v1   v2c) security-level (authentication   none   privacy)]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Associate the view with a community or a group name (SNMPv3).
<b>Options</b>	<i>view-name</i> —The name of the view to which the SNMP user group has access.
<b>Usage Guidelines</b>	See “Configuring the Read View” on page 65.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	Configuring MIB Views on page 45

## 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 77.

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

## routing-instance

---

<b>Syntax</b>	<code>routing-instance <i>routing-instance-name</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
<b>Release Information</b>	Statement introduced in JUNOS Release 8.3.
<b>Description</b>	Specify a routing instance for an SNMPv3 trap target.
<b>Options</b>	<p><i>routing-instance-name</i>—Name of the routing instance.</p> <p>To configure a routing instance within a logical system, specify the logical system name followed by the routing instance name. Use a slash ( / ) to separate the two names (for example, <code>test-ls/test-ri</code>). To configure the default routing instance on a logical system, specify the logical system name followed by <b>default</b> (for example, <code>test-ls/default</code>).</p>
<b>Usage Guidelines</b>	See “Configuring the Trap Target Address” on page 70.
<b>Required Privilege Level</b>	<p><code>snmp</code>—To view this statement in the configuration.</p> <p><code>snmp-control</code>—To add this statement to the configuration.</p>

## security-level

---

See the following sections:

- security-level (Defining Access Privileges) on page 202
- security-level (Generating SNMP Notifications) on page 202

### **security-level (Defining Access Privileges)**

<b>Syntax</b>	security-level (authentication   none   privacy);
<b>Hierarchy Level</b>	[edit snmp v3 vacm access group group-name default-context-prefix security-model (any   usm   v1   v2c)]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define the security level used for access privileges.
<b>Options</b>	<p>none—No authentication and no encryption.</p> <p>authentication—Provides authentication but no encryption.</p> <p>privacy—Provides authentication and encryption.</p> <p><b>Default:</b> none</p>
<b>Usage Guidelines</b>	See “Configuring the Security Level” on page 63.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

### **security-level (Generating SNMP Notifications)**

<b>Syntax</b>	security-level (authentication   none   privacy);
<b>Hierarchy Level</b>	[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the security level to use when generating SNMP notifications.
<b>Options</b>	<p>none—No authentication and no encryption.</p> <p>authentication—Provides authentication but no encryption.</p> <p>privacy—Provides authentication and encryption.</p> <p><b>Default:</b> none</p>
<b>Usage Guidelines</b>	See “Configuring the Security Level” on page 75.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>



## security-model

---

See the following sections:

- security-model (Access Privileges) on page 203
- security-model (Group) on page 203
- security-model (SNMP Notifications) on page 204

### **security-model (Access Privileges)**

<b>Syntax</b>	security-model (usm   v1   v2c);
<b>Hierarchy Level</b>	[edit snmp v3 vacm access group <i>group-name</i> default-context-prefix]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure a group's security model used for access privileges.
<b>Options</b>	usm—SNMPv3 security model.  v1—SNMPv1 security model.  v2c—SNMPv2c security model.
<b>Usage Guidelines</b>	See “Configuring the Security Model” on page 63.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

### **security-model (Group)**

<b>Syntax</b>	security-model (usm   v1   v2c);
<b>Hierarchy Level</b>	[edit snmp v3 vacm security-to-group]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Define a security model for a group.
<b>Options</b>	usm—SNMPv3 security model.  v1—SNMPv1 security model.  v2c—SNMPv2c security model.
<b>Usage Guidelines</b>	See “Configuring the Security Model” on page 66.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

**security-model (SNMP Notifications)**

<b>Syntax</b>	security-model (usm   v1   v2c);
<b>Hierarchy Level</b>	[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure a group's security model used with sending notifications.
<b>Options</b>	usm—SNMPv3 security model.  v1—SNMPv1 security model.  v2c—SNMPv2c security model.
<b>Usage Guidelines</b>	See “Configuring the Security Model” on page 75.
<b>Required Privilege Level</b>	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 205
- security-name (Security Group) on page 206
- security-name (SNMP Notifications) on page 206

### 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*—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 you configure traps or informs.

---


**Usage Guidelines** See “Configuring the Security Names” on page 81.

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

**security-name (Security Group)**

<b>Syntax</b>	<code>security-name security-name;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 vacm security-to-group security-model (usm   v1   v2c)]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Associate a group or a community string with a configured security group.
<b>Options</b>	<i>security-name</i> —Username configured at the <code>[edit snmp v3 usm local-engine user username]</code> hierarchy level. For SNMPv1 and SNMPv2c, the security name is the community string configured at the <code>[edit snmp v3 snmp-community community-index]</code> hierarchy level.
<b>Usage Guidelines</b>	See “Configuring the Security Name” on page 67.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

**security-name (SNMP Notifications)**

<b>Syntax</b>	<code>security-name security-name;</code>
<b>Hierarchy Level</b>	<code>[edit snmp v3 target-parameters target-parameters-name parameters]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the security name used when generating SNMP notifications.
<b>Options</b>	<i>security-name</i> —Identifies the user that is used when generating the notification if the USM security model is used. Identifies the SNMP community used when generating the notification if the v1 or v2c security models are used.
<hr/> <div>  <b>NOTE:</b> The access privileges for the group associated with this security name must allow this notification to be sent. </div> <p>If you are using the v1 or v2 security models, the security name at the <code>[edit snmp v3 vacm security-to-group]</code> hierarchy level must match the security name at the <code>[edit snmp v3 snmp-community community-index]</code> hierarchy level.</p> <hr/>	
<b>Usage Guidelines</b>	See “Configuring the Security Name” on page 76.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

## security-to-group

---

<b>Syntax</b>	<pre>security-to-group {   security-model (usm   v1   v2c) {     security-name <i>security-name</i>;     group <i>group-name</i>;   } }</pre>
<b>Hierarchy Level</b>	[edit snmp v3 vacm]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	<p>Configure the group to which a specific security name belongs.</p> <p>The remaining statements are explained separately.</p>
<b>Usage Guidelines</b>	See “Assigning Security Names to Groups” on page 66.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## snmp-community

---

<b>Syntax</b>	<pre>snmp-community <i>community-index</i> {   community-name <i>community-name</i>;   security-name <i>security-name</i>;   tag <i>tag-name</i>; }</pre>
<b>Hierarchy Level</b>	[edit snmp v3]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the SNMP community.
<b>Options</b>	<p><i>community-index</i>—(Optional) String that identifies an SNMP community.</p> <p>The remaining statements are explained separately.</p>
<b>Usage Guidelines</b>	See “Configuring the SNMPv3 Community” on page 80.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## tag

---

<b>Syntax</b>	<code>tag tag-name;</code>
<b>Hierarchy Level</b>	[edit snmp v3 notify <i>name</i> , [edit snmp v3 snmp-community <i>community-index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure a set of targets to receive traps or informs (for IPv4 packets only).
<b>Options</b>	<i>tag-name</i> —Identifies the address of managers that are allowed to use a community string.
<b>Usage Guidelines</b>	See “Configuring the Tag” on page 82 and “Configuring the SNMPv3 Trap Notification” on page 69.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## tag-list

---

<b>Syntax</b>	<code>tag-list tag-list;</code>
<b>Hierarchy Level</b>	[edit snmp v3 target-address <i>target-address-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure an SNMP tag list used to select target addresses.
<b>Options</b>	<i>tag-list</i> —Defines sets of target addresses. To specify more than one tag, specify the tag names as a space-separated list enclosed within double quotes.
<b>Usage Guidelines</b>	See “Configuring the Tag List” on page 72.
<b>Required Privilege Level</b>	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;  
     routing-instance instance;  
     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*—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 70.

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

## target-parameters

---

<b>Syntax</b>	target-parameters <i>target-parameters-name</i> { notify-filter <i>profile-name</i> ; parameters { message-processing-model (v1   v2c   V3); security-model ( usm   v1   v2c); security-level (authentication   none   privacy); security-name <i>security-name</i> ; } }
<b>Hierarchy Level</b>	[edit snmp v3]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure the message processing and security parameters to be used in sending notifications to a particular management target.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Defining and Configuring the Trap Target Parameters” on page 74.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## type

---

<b>Syntax</b>	type (trap   inform);
<b>Hierarchy Level</b>	[edit snmp v3 notify <i>name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4. inform option added in JUNOS Release 7.4.
<b>Description</b>	Configure the type of notification.
<b>Options</b>	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.
<b>Usage Guidelines</b>	See “Configuring the SNMPv3 Trap Notification” on page 69 and “Configuring SNMP Informs” on page 76.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.



**user**

---

<b>Syntax</b>	<code>user <i>username</i>;</code>
<b>Hierarchy Level</b>	[edit snmp v3 usm local-engine]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify a user associated with an SNMPv3 group.
<b>Options</b>	<i>username</i> —SNMPv3 USM username.
<b>Usage Guidelines</b>	See “Creating SNMPv3 Users” on page 57.
<b>Required Privilege Level</b>	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;
                    }
                }
            }
            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]

<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure user-based security model (USM) information.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Creating SNMPv3 Users” on page 57.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

**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;
        routing-instance instance;
        tag-list tag-list;
        target-parameters target-parameters-name;
    }
    target-parameters target-parameters-name {
        notify-filter profile-name;
        parameters {
            message-processing-model (v1 | v2c | V3);
            security-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;
    }
}
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;
        }
    }
}
}
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 53.

**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 62.

**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 45.

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

**Related Topics**    Associating MIB Views with an SNMP User Group on page 64

## write-view

---

<b>Syntax</b>	<code>write-view view-name;</code>
<b>Hierarchy Level</b>	[edit snmp v3 vacm access group <i>group-name</i> default-context-prefix security-model (any   usm   v1   v2c) security-level (authentication   none   privacy)]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Associate the view with a community or a group name (SNMPv3).
<b>Options</b>	<i>view-name</i> —The name of the view to which the SNMP user group has access.
<b>Usage Guidelines</b>	See “Configuring MIB Views” on page 45.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	Configuring MIB Views on page 45



## **Part 4**

# **RMON Alarms and Events**

- Configuring RMON Alarms and Events on page 221
- Monitoring RMON Alarms and Events on page 229
- Summary of RMON Alarm and Event Configuration Statements on page 239



## Chapter 15

# Configuring RMON Alarms and Events

This chapter contains the following topics:

- Understanding RMON Alarms and Events Configuration on page 221
- Configuring an Alarm Entry and Its Attributes on page 222
- Configuring an Event Entry and Its Attributes on page 226
- Example: Configuring an RMON Alarm and Event Entry on page 227

### Understanding RMON Alarms and Events Configuration

---

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 221 and “Summary of RMON Alarm and Event Configuration Statements” on page 239.

For more information on monitoring integer-valued MIB objects, see “Monitoring RMON Alarms and Events” on page 229.

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;
    falling-threshold-interval seconds;
    interval seconds;
    rising-event-index index;
    rising-threshold integer;
    request-type (get-next-request | get-request | walk-request);
    sample-type (absolute-value | delta-value);
    startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);
    syslog-subtag syslog-subtag;
    variable oid-variable;
    event index {
```

```

        community community-name;
        description description;
        type type;
    }
}

```

This topic also describes the minimum required RMON alarm and event configuration:

- Minimum RMON Alarm and Event Entry Configuration on page 222

### **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 on page 223
- Configuring the Description on page 223
- Configuring the Falling Event Index or Rising Event Index on page 223
- Configuring the Falling Threshold or Rising Threshold on page 224
- Configuring the Interval on page 224
- Configuring the Falling Threshold Interval on page 224
- Configuring the Request Type on page 225
- Configuring the Sample Type on page 225
- Configuring the Startup Alarm on page 226
- Configuring the System Log Tag on page 226
- Configuring the Variable on page 226

## 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;
    falling-threshold-interval seconds;
    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.

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 Falling Threshold Interval

The falling threshold interval represents the interval between samples when the rising threshold is crossed. Once the alarm crosses the falling threshold, the regular sampling interval is used.



**NOTE:** You cannot configure the falling threshold interval for alarms that have the request type set to `walk-request`.

To configure the falling threshold interval, include the `falling-threshold interval` statement at the `[edit snmp rmon alarm index]` hierarchy level and specify the number of seconds:

```
[edit snmp rmon alarm index]
  falling-threshold-interval seconds;
```

`seconds` can be a value from 1 through 2,147,483,647. The default is 60 seconds.

### Configuring the Request Type

By default an RMON alarm can monitor only one object instance (as specified in the configuration). You can configure a `request-type` statement to extend the scope of the RMON alarm to include all object instances belonging to a MIB branch or to include the next object instance after the instance specified in the configuration.

To configure the request type, include the `request-type` statement at the `[edit snmp rmon alarm index]` hierarchy level and specify `get-next-request`, `get-request`, or `walk-request`:

```
[edit snmp rmon alarm index]
  request-type (get-next-request | get-request | walk-request);
```

`walk` extends the RMON alarm configuration to all object instances belonging to a MIB branch. `next` extends the RMON alarm configuration to include the next object instance after the instance specified in the configuration.

### 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 System Log Tag

The **syslog-subtag** statement specifies the tag to be added to the system log message. You can specify a string of not more than 80 uppercase characters as the system log tag.

To configure the system log tag, include the **syslog-subtag** statement at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
syslog-subtag syslog-subtag;
```

## 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 16

# 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 routing platform. 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 on page 229
- Using `alarmTable` to Monitor MIB Objects on page 231
- RMON Events on page 234

## 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 topic covers the following sections:

- `alarmTable` on page 230
- `jnxRmonAlarmTable` on page 230

## ***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`—Method for adding and removing 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 Events and Alarms and Event MIB, see

[www.juniper.net/techpubs/software/junos922/swconfig-net-mgmt/mib-jnx-rmon.txt](http://www.juniper.net/techpubs/software/junos922/swconfig-net-mgmt/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 407.

## Using alarmTable to Monitor MIB Objects

---

To use alarmTable to monitor a MIB object, perform the following tasks:

- Creating an Alarm Entry on page 231
- Configuring the Alarm MIB Objects on page 231
- Activating a New Row in alarmTable on page 234
- Modifying an Active Row in alarmTable on page 234
- Deactivating a Row in alarmTable on page 234

### 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 on page 232
- alarmVariable on page 232
- alarmSampleType on page 232
- alarmValue on page 232
- alarmStartupAlarm on page 232
- alarmRisingThreshold on page 233
- alarmFallingThreshold on page 233
- alarmOwner on page 233
- alarmRisingEventIndex on page 233
- alarmFallingEventIndex on page 233

### 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` on page 234
- Using `eventTable` to Log Alarms on page 235

### ***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 235
- Configuring the MIB Objects on page 235
- Activating a New Row in **eventTable** on page 237
- Deactivating a Row in **eventTable** on page 237

### 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 236
- **eventCommunity** on page 236
- **eventOwner** on page 236
- **eventDescription** on page 237

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 a 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 17

# 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*;  
              falling-threshold-interval *seconds*;  
              interval *seconds*;  
              rising-event-index *index*;  
              rising-threshold *integer*;  
              request-type (get-next-request | get-request | walk-request);  
              sample-type (absolute-value | delta-value);  
              startup-alarm (falling-alarm | rising-alarm | rising-or-falling alarm);  
              syslog-subtag *syslog-subtag*;  
              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 222.

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

**Related Topics**    event

## community

---

<b>Syntax</b>	<code>community <i>community-name</i>;</code>
<b>Hierarchy Level</b>	[edit snmp rmon event <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	The trap group that is used when generating a trap (if <b>eventType</b> is configured to send traps). If that trap group has the <b>rmon-alarm</b> 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 <b>eventCommunity</b> ). If nothing is configured, traps are sent to each group with the <b>rmon-alarm</b> category set.
<b>Options</b>	<i>community-name</i> —Identifies the trap group that is used when generating a trap if the event is configured to send traps.
<b>Usage Guidelines</b>	See “Configuring an Event Entry and Its Attributes” on page 226.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## description

---

<b>Syntax</b>	<code>description <i>description</i>;</code>
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ], [edit snmp rmon event <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Text description of alarm or event.
<b>Options</b>	<i>description</i> —Text description of an alarm or event entry. If the description includes spaces, enclose it in quotation marks (" ").
<b>Usage Guidelines</b>	See “Configuring the Description” on page 223 and “Configuring an Event Entry and Its Attributes” on page 226.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

**event**

---

<b>Syntax</b>	event <i>index</i> { community <i>community-name</i> ; description <i>description</i> ; type <i>type</i> ; }
<b>Hierarchy Level</b>	[edit snmp rmon]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure RMON event entries.
<b>Options</b>	<i>index</i> —Identifier for a specific event entry.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Configuring an Event Entry and Its Attributes” on page 226.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	alarm

**falling-event-index**

---

<b>Syntax</b>	falling-event-index <i>index</i> ;
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	The index of the event entry that is used when a falling threshold is crossed. If this value is zero, no event is triggered.
<b>Options</b>	<i>index</i> —Index of the event entry that is used when a falling threshold is crossed. <b>Range:</b> 0 through 65,535 <b>Default:</b> 0
<b>Usage Guidelines</b>	See “Configuring the Falling Event Index or Rising Event Index” on page 223.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	rising-event-index

## falling-threshold

---

<b>Syntax</b>	<code>falling-threshold <i>integer</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp rmon alarm <i>index</i>]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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 <code>startup-alarm</code> is equal to <code>falling-alarm</code> or <code>rising-or-falling-alarm</code> . After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the <code>rising-threshold</code> .
<b>Options</b>	<i>integer</i> —The lower threshold for the alarm entry. <b>Range:</b> -2,147,483,648 through 2,147,483,647 <b>Default:</b> 20 percent less than <code>rising-threshold</code>
<b>Usage Guidelines</b>	See “Configuring the Falling Threshold or Rising Threshold” on page 224.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
<b>Related Topics</b>	<code>rising-threshold</code>

## falling-threshold-interval

---

<b>Syntax</b>	<code>falling-threshold-interval <i>seconds</i>;</code>
<b>Hierarchy Level</b>	<code>[edit snmp rmon alarm <i>index</i>]</code>
<b>Release Information</b>	Statement introduced in JUNOS Release 8.3.
<b>Description</b>	Interval between samples when the rising threshold is crossed. Once the alarm crosses the falling threshold, the regular sampling interval is used.
<b>Options</b>	<i>interval</i> —Time between samples, in seconds. <b>Range:</b> 1 through 2,147,483,647 seconds <b>Default:</b> 60 seconds
<b>Usage Guidelines</b>	See “Configuring the Falling Threshold Interval” on page 224.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
<b>Related Topics</b>	<code>interval</code>



## interval

---

<b>Syntax</b>	interval <i>seconds</i> ;
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Interval between samples.
<b>Options</b>	<i>interval</i> —Time between samples, in seconds. <b>Range:</b> 1 through 2,147,483,647 seconds <b>Default:</b> 60 seconds
<b>Usage Guidelines</b>	See “Configuring the Interval” on page 224.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## request-type

---

<b>Syntax</b>	request-type (get-next-request   get-request   walk-request);
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.3.
<b>Description</b>	Extends monitoring to a specific SNMP object instance ( <b>get-request</b> ), or extends monitoring to all object instances belonging to a MIB branch ( <b>walk-request</b> ), or extends monitoring to the next object instance after the instance specified in the configuration ( <b>get-next-request</b> ).
<b>Options</b>	get-next-request—Performs an SNMP get next request.  get-request—Performs an SNMP get request.  walk-request—Performs an SNMP walk request. <b>Default:</b> walk-request
<b>Usage Guidelines</b>	See “Configuring the Request Type” on page 225.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	variable

## rising-event-index

---

<b>Syntax</b>	<code>rising-event-index index;</code>
<b>Hierarchy Level</b>	<code>[edit snmp rmon alarm index]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	The index of the event entry that is used when a rising threshold is crossed. If this value is zero, no event is triggered.
<b>Options</b>	<i>index</i> —Index of the event entry that is used when a rising threshold is crossed. <b>Range:</b> 0 through 65,535 <b>Default:</b> 0
<b>Usage Guidelines</b>	See “Configuring the Falling Event Index or Rising Event Index” on page 223.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	falling-event-index

## rising-threshold

---

<b>Syntax</b>	<code>rising-threshold integer;</code>
<b>Hierarchy Level</b>	<code>[edit snmp rmon alarm index]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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.
<b>Options</b>	<i>integer</i> —The lower threshold for the alarm entry. <b>Range:</b> -2,147,483,648 through 2,147,483,647
<b>Usage Guidelines</b>	See “Configuring the Falling Threshold or Rising Threshold” on page 224.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	falling-threshold

**rmon**

---

<b>Syntax</b>	<code>rmon { ... }</code> <code>}</code>
<b>Hierarchy Level</b>	<code>[edit snmp]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure Remote Monitoring.
<b>Usage Guidelines</b>	See “Configuring RMON Alarms and Events” on page 221.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

**sample-type**

---

<b>Syntax</b>	<code>sample-type (absolute-value   delta-value);</code>
<b>Hierarchy Level</b>	<code>[edit snmp rmon alarm index]</code>
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Method of sampling the selected variable.
<b>Options</b>	<p><code>absolute-value</code>—Actual value of the selected variable is used when comparing against the thresholds.</p> <p><code>delta-value</code>—Difference between samples of the selected variable is used when comparing against the thresholds.</p>
<b>Usage Guidelines</b>	See “Configuring the Sample Type” on page 225.
<b>Required Privilege Level</b>	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.

## startup-alarm

---

<b>Syntax</b>	startup-alarm (falling-alarm   rising-alarm   rising-or-falling-alarm);
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	The alarm that can be sent upon entry startup.
<b>Options</b>	<p>falling-alarm—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.</p> <p>rising-alarm—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.</p> <p>rising-or-falling-alarm—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.</p> <p><b>Default:</b> rising-or-falling-alarm</p>
<b>Usage Guidelines</b>	See “Configuring the Startup Alarm” on page 226.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## syslog-subtag

---

<b>Syntax</b>	syslog-subtag <i>syslog-subtag</i> ;
<b>Hierarchy Level</b>	[edit snmp rmon event <i>index</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.5.
<b>Description</b>	Tag to be added to the system log message. The syslog-subtag can be a string of not more than 80 uppercase characters.
<b>Usage Guidelines</b>	See “Configuring the System Log Tag” on page 226.
<b>Required Privilege Level</b>	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>

## type

---

<b>Syntax</b>	<code>type type;</code>
<b>Hierarchy Level</b>	[edit snmp rmon event <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Type of notification generated when a threshold is crossed.
<b>Options</b>	<p><i>type</i>—Type of notification. It can be one of the following:</p> <ul style="list-style-type: none"> <li>■ <code>log</code>—Add an entry to logTable.</li> <li>■ <code>log-and-trap</code>—Send an SNMP trap and make a log entry.</li> <li>■ <code>none</code>—No notifications are sent.</li> <li>■ <code>snmptrap</code>—Send an SNMP trap.</li> </ul> <p><b>Default:</b> <code>log-and-trap</code></p>
<b>Usage Guidelines</b>	See “Configuring an Event Entry and Its Attributes” on page 226.
<b>Required Privilege Level</b>	<p><code>snmp</code>—To view this statement in the configuration.</p> <p><code>snmp-control</code>—To add this statement to the configuration.</p>

## variable

---

<b>Syntax</b>	<code>variable oid-variable;</code>
<b>Hierarchy Level</b>	[edit snmp rmon alarm <i>index</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Object identifier (OID) of MIB variable to be monitored.
<b>Options</b>	<i>oid-variable</i> —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).
<b>Usage Guidelines</b>	See “Configuring the Variable” on page 226.
<b>Required Privilege Level</b>	<p><code>snmp</code>—To view this statement in the configuration.</p> <p><code>snmp-control</code>—To add this statement to the configuration.</p>



## **Part 5**

# **Health Monitoring**

- Configuring Health Monitoring on page 251
- Summary of Health Monitoring Configuration Statements on page 257





## Chapter 18

# Configuring Health Monitoring

This chapter contains the following topics:

- Configuring Health Monitoring on JUNOS Devices on page 251
- Example: Configuring Health Monitoring on page 254

### Configuring Health Monitoring on JUNOS Devices

---

As the number of devices managed by a typical network management system (NMS) grows and the complexity of the devices themselves increases, it becomes increasingly impractical for the NMS to use polling to monitor the devices. A more scalable approach is to rely on network devices to notify the NMS when something requires attention.

On Juniper Networks routing platforms, RMON alarms and events provide much of the infrastructure needed to reduce the polling overhead from the NMS. (For more information, see “Configuring RMON Alarms and Events” on page 221.) However, with this approach, you must set up the NMS to configure specific MIB objects into RMON alarms. This often requires device-specific expertise and customizing of the monitoring application. In addition, some MIB object instances that need monitoring are set only at initialization or change at runtime and cannot be configured in advance.

To address these issues, the health monitor extends the RMON alarm infrastructure to provide predefined monitoring for a selected set of object instances (for file system usage, CPU usage, and memory usage) and includes support for unknown or dynamic object instances (such as JUNOS processes).

Health monitoring is designed to minimize user configuration requirements. To configure health monitoring entries, you include statements at the `[edit snmp]` hierarchy level of the configuration:

```
[edit snmp]
health-monitor {
    falling-threshold percentage;
    interval seconds;
    rising-threshold percentage;
}
```

You can use the `show snmp health-monitor` operational command to view information about health monitor alarms and logs.

As the number of devices managed by a typical network management system (NMS) grows and the complexity of the devices themselves increases, it becomes increasingly impractical for the NMS to use polling to monitor the devices. A more scalable approach is to rely on network devices to notify the NMS when something requires attention.

On Juniper Networks routing platforms, RMON alarms and events provide much of the infrastructure needed to reduce the polling overhead from the NMS. (For more information, see “Configuring RMON Alarms and Events” on page 221.) However, with this approach, you must set up the NMS to configure specific MIB objects into RMON alarms. This often requires device-specific expertise and customizing of the monitoring application. In addition, some MIB object instances that need monitoring are set only at initialization or change at runtime and cannot be configured in advance.

To address these issues, the health monitor extends the RMON alarm infrastructure to provide predefined monitoring for a selected set of object instances (for file system usage, CPU usage, and memory usage) and includes support for unknown or dynamic object instances (such as JUNOS processes).

Health monitoring is designed to minimize user configuration requirements. To configure health monitoring entries, you include statements at the `[edit snmp]` hierarchy level of the configuration:

```
[edit snmp]
health-monitor {
    falling-threshold percentage;
    interval seconds;
    rising-threshold percentage;
}
```

You can use the `show snmp health-monitor` operational command to view information about health monitor alarms and logs.

This topic describes the minimum required configuration and discusses the following tasks for configuring the health monitor:

- Monitored Objects on page 252
- Minimum Health Monitoring Configuration on page 253
- Configuring the Falling Threshold or Rising Threshold on page 253
- Configuring the Interval on page 254
- Log Entries and Traps on page 254

## Monitored Objects

When you configure the health monitor, monitoring information for certain object instances is available, as shown in Table 25 on page 253.

**Table 25: Monitored Object Instances**

Object	Description
jnxHrStoragePercentUsed.1	Monitors the following file system on the router:  /dev/ad0s1a:  This is the root file system mounted on /.
jnxHrStoragePercentUsed.2	Monitors the following file system on the router:  /dev/ad0s1e:  This is the configuration file system mounted on /config
jnxOperatingCPU (RE0)	Monitors CPU usage for Routing Engines (RE0 and RE1). The index values assigned to Routing Engines depend on whether the Chassis MIB uses a zero-based or ones-based indexing scheme. Because the indexing scheme is configurable, the proper index is determined when the router is initialized and when there is a configuration change. If the router has only one Routing Engine, the alarm entry monitoring RE1 is removed after five failed attempts to obtain the CPU value.
jnxOperatingCPU (RE1)	
jnxOperatingBuffer (RE0)	Monitors the amount of memory available on Routing Engines (RE0 and RE1). Because the indexing of this object is identical to that used for jnxOperatingCPU, index values are adjusted depending on the indexing scheme used in the Chassis MIB. As with jnxOperatingCPU, the alarm entry monitoring RE1 is removed if the router has only one Routing Engine.
jnxOperatingBuffer (RE1)	
sysAppElmtRunCPU	Monitors the CPU usage for each JUNOS process (also called daemon). Multiple instances of the same process are monitored and indexed separately.
sysAppElmtRunMemory	Monitors the memory usage for each JUNOS process. Multiple instances of the same process are monitored and indexed separately.

### Minimum Health Monitoring Configuration

To enable health monitoring on the router, include the **health-monitor** statement at the [edit snmp] hierarchy level:

```
[edit snmp]
health-monitor;
```

### Configuring the Falling Threshold or Rising Threshold

The falling threshold is the lower threshold (expressed as a percentage of the maximum possible value) 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. 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 a percentage of the maximum possible value. The default is 70 percent.

By default, the rising threshold is 80 percent of the maximum possible value for the monitored object instance. 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. 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 a percentage of the maximum possible value for the monitored variable.

To configure the falling threshold or rising threshold, include the `falling-threshold` or `rising-threshold` statement at the `[edit snmp health-monitor]` hierarchy level:

```
[edit snmp health-monitor]
  falling-threshold percentage;
  rising-threshold percentage;
```

*percentage* can be a value from 1 through 100.

The falling and rising thresholds apply to all object instances monitored by the health monitor.

## Configuring the Interval

The interval represents the period of time, in seconds, over which the object instance 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 health-monitor]` hierarchy level:

```
[edit snmp health-monitor]
  interval seconds;
```

*seconds* can be a value from 1 through 2147483647. The default is 300 seconds (5 minutes).

## Log Entries and Traps

The system log entries generated for any health monitor events (thresholds crossed, errors, and so on) have a corresponding `HEALTHMONITOR` tag rather than a generic `SNMPD_RMON_EVENTLOG` tag. However, the health monitor sends generic `RMON risingThreshold` and `fallingThreshold` traps.

## Example: Configuring Health Monitoring

---

Configure the health monitor:

```
[edit snmp]
health-monitor {
    falling-threshold 85;
    interval 600;
    rising-threshold 75;
}
```

In this example, the sampling interval is every **600** seconds (10 minutes), the falling threshold is **85** percent of the maximum possible value for each object instance monitored, and the rising threshold is **75** percent of the maximum possible value for each object instance monitored.



## Chapter 19

# Summary of Health Monitoring Configuration Statements

The following sections explain each of the health monitoring configuration statements. The statements are organized alphabetically.

### falling-threshold

---

<b>Syntax</b>	falling-threshold <i>percentage</i> ;
<b>Hierarchy Level</b>	[edit snmp health-monitor]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.0.
<b>Description</b>	The lower threshold is expressed as a percentage of the maximum possible value 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. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the <b>rising-threshold</b> .
<b>Options</b>	<i>percentage</i> —The lower threshold for the alarm entry. <b>Range:</b> 1 through 100 <b>Default:</b> 70 percent of the maximum possible value
<b>Usage Guidelines</b>	See “Configuring the Falling Threshold or Rising Threshold” on page 253.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	rising-threshold

## health-monitor

---

<b>Syntax</b>	health-monitor{ falling-threshold <i>percentage</i> ; interval <i>seconds</i> ; rising-threshold <i>percentage</i> ; }
<b>Hierarchy Level</b>	[edit snmp]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.0.
<b>Description</b>	Configure health monitoring.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Configuring Health Monitoring” on page 251.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.

## interval

---

<b>Syntax</b>	interval <i>seconds</i> ;
<b>Hierarchy Level</b>	[edit snmp health-monitor]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.0.
<b>Description</b>	Interval between samples.
<b>Options</b>	<i>interval</i> —Time between samples, in seconds. <b>Range:</b> 1 through 2147483647 seconds <b>Default:</b> 300 seconds
<b>Usage Guidelines</b>	See “Configuring the Interval” on page 254.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.



## rising-threshold

---

<b>Syntax</b>	rising-threshold <i>percentage</i> ;
<b>Hierarchy Level</b>	[edit snmp health-monitor]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.0.
<b>Description</b>	The upper threshold is expressed as a percentage of the maximum possible value 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. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the <b>falling-threshold</b> .
<b>Options</b>	<i>integer</i> —The lower threshold for the alarm entry. <b>Range:</b> 1 through 100 <b>Default:</b> 80 percent of the maximum possible value
<b>Usage Guidelines</b>	See “Configuring the Falling Threshold or Rising Threshold” on page 253.
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
<b>Related Topics</b>	falling-threshold



## **Part 6**

# **Monitoring Service Quality**

- Monitoring Service Quality in Service Provider Networks on page 263



## Chapter 20

# 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:

- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 263
- Understanding RMON for Monitoring Service Quality on page 265
- Defining and Measuring Network Availability on page 269
- Measuring Health on page 274
- Measuring Performance on page 280

## Understanding Measurement Points, Key Performance Indicators, and Baseline Values

---

This topic contains the following sections:

- Measurement Points on page 263
- Basic Key Performance Indicators on page 264
- Setting Baselines on page 265

### **Measurement Points**

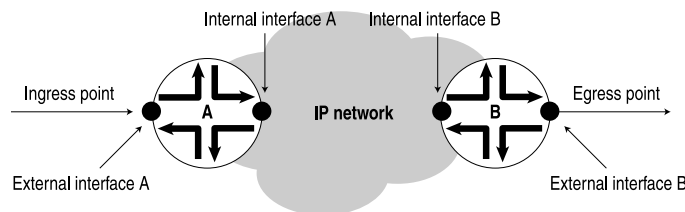
Defining the measurement points where metrics are measured is equally as important as defining the metrics themselves. This section describes measurement points within the context of this chapter and helps identify where measurements can be taken from a service provider network. It is important to understand 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 by physical links that are all running the Internet Protocol. You can view the network as a collection of routers with an ingress (entry) point and an egress (exit) point. See Figure 3 on page 264.

- 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 ensure that the correct router subcomponents have been identified in advance.

**Figure 3: Network Entry Points**



9017042



**NOTE:** Figure 3 on page 264 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).

## 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.

## Understanding RMON for Monitoring Service Quality

---

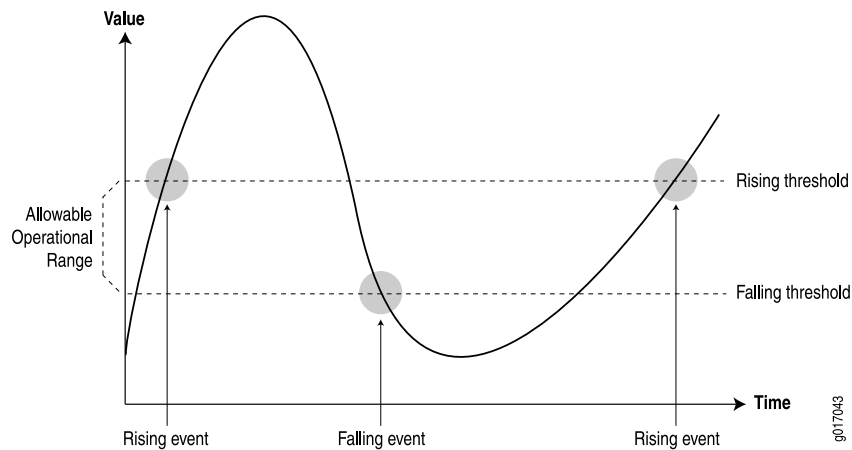
Health and performance monitoring can benefit from the remote monitoring of SNMP variables by the local SNMP agents running on each router. The SNMP agents compare MIB values against predefined thresholds and generate exception alarms without the need for polling by a central SNMP management platform. 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 topic includes the following sections:

- Setting Thresholds on page 265
- RMON Command-Line Interface on page 266
- RMON Event Table on page 267
- RMON Alarm Table on page 267
- Troubleshooting RMON on page 268

## 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 4 on page 266.)

**Figure 4: 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, include the following configuration statements at the `[edit snmp]` hierarchy level:

```
rmon {
  alarm index {
    description;
    falling-event-index;
```



```

    falling-threshold;
    intervals;
    rising-event-index;
    rising-threshold;
    sample-type (absolute-value | delta-value);
    startup-alarm (falling | rising | rising-or-falling);
    variable;
}
event index {
    community;
    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 26 on page 267.) To configure RMON using SNMP, perform SNMP **Set** requests 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 the events have been configured, you do not need to update them.

**Table 26: RMON Event Table**

Field	Description
eventDescription	Text description of this event
eventType	Type of event (for example, log, trap, or log and trap)
eventCommunity	Trap group to which to send this event (as defined in the JUNOS software configuration, which is not the same as the community)
eventOwner	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 27 on page 268.

**Table 27: RMON Alarm Table**

Field	Description
alarmStatus	Status of this row (for example, <code>valid</code> , <code>invalid</code> , or <code>createRequest</code> )
alarmInterval	Sampling period (in seconds) of the monitored variable
alarmVariable	OID (and instance) of the variable to be monitored
alarmValue	Actual value of the sampled variable
alarmSampleType	Sample type ( <code>absolute</code> or <code>delta</code> changes)
alarmStartupAlarm	Initial alarm ( <code>rising</code> , <code>falling</code> , or <code>either</code> )
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 28 on page 268 to the RFC 2819 `alarmTable`.

**Table 28: jnxRmon Alarm Extensions**

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

Monitoring the extensions in this table provides clues as to why remote alarms may be not behave as expected.

## Defining and Measuring Network Availability

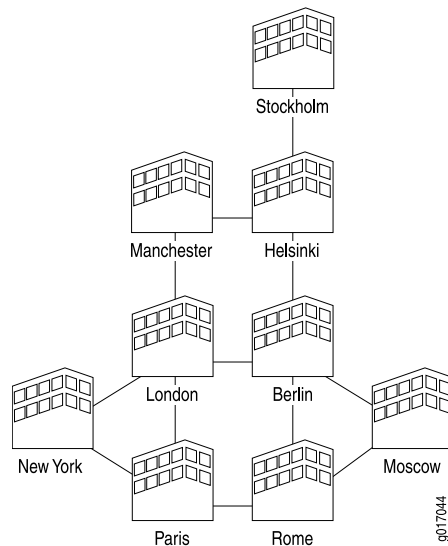
This topic covers the following sections:

- Defining Network Availability on page 269
- Measuring Availability on page 271

### Defining Network Availability

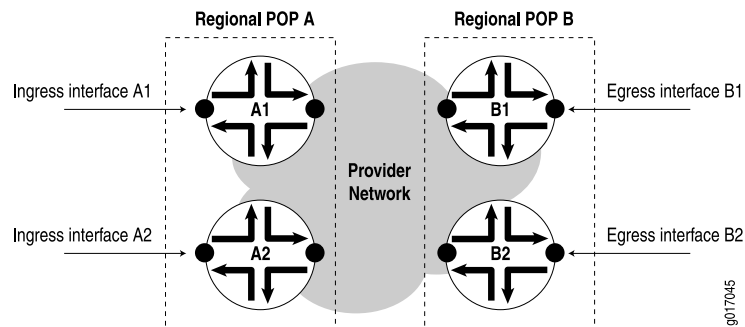
Availability of a service provider's IP network can be thought of as the reachability between the regional points of presence (POP), as shown in Figure 5 on page 269.

**Figure 5: 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 its 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 6 on page 270.

**Figure 6: Measurements to Each Router**

Measurements include:

- Path availability—Availability of an egress interface B1 as seen from an ingress interface A1.
- Router availability—Percentage of path availability of all measured paths terminating on the router.
- POP availability—Percentage of router availability between any two regional POPs, A and B.
- Network availability—Percentage of POP availability for all regional POPs in the service provider's network.

To measure POP availability of POP A to POP B in Figure 6 on page 270, 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 significant management traffic. From the sample 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:

$$[n \times (n-1)] / 2 \text{ gives } [68 \times (68-1)] / 2 = 2278 \text{ paths}$$

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:

$$[n \times (n-1)] / 2 \text{ gives } [24 \times (24-1)] / 2 = 276 \text{ measurements}$$

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 Mbps, a utilization of 1.362 percent
- On an OC12/STM4 link of 622.08 Mbps, 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. With one test per minute to each destination, a total of  $1 \times 60 \times 24 \times 276 = 397,440$  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 real-time performance monitoring as a proactive monitoring solution.

## Real-Time Performance Monitoring

Juniper Networks provides a real-time performance monitoring (RPM) service to monitor real-time network performance. Use the J-Web Quick Configuration feature to configure real-time performance monitoring parameters used in real-time performance monitoring tests. (J-Web Quick Configuration is a browser-based GUI that runs on Juniper Networks routers. For more information, see the *J-Web Interface User Guide*.)

### Configuring Real-Time Performance Monitoring

Some of the most common options you can configure for real-time performance monitoring tests are shown in Table 29 on page 272.

**Table 29: Real-Time Performance Monitoring Configuration Options**

Field	Description
<b>Request Information</b>	
Probe Type	Type of probe to send as part of the test. Probe types can be: <ul style="list-style-type: none"> <li>■ http-get</li> <li>■ http-get-metadata</li> <li>■ icmp-ping</li> <li>■ icmp-ping-timestamp</li> <li>■ tcp-ping</li> <li>■ udp-ping</li> </ul>
Interval	Wait time (in seconds) between each probe transmission. The range is 1 to 255 seconds.
Test Interval	Wait time (in seconds) between tests. The range is 0 to 86400 seconds.
Probe Count	Total number of probes sent for each test. The range is 1 to 15 probes.
Destination Port	TCP or UDP port to which probes are sent. Use number 7—a standard TCP or UDP port number—or select a port number from 49152 through 65535.
DSCP Bits	Differentiated Services code point (DSCP) bits. This value must be a valid 6-bit pattern. The default is 000000.
Data Size	Size (in bytes) of the data portion of the ICMP probes. The range is 0 to 65507 bytes.
Data Fill	Contents of the data portion of the ICMP probes. Contents must be a hexadecimal value. The range is 1 to 800h.
<b>Maximum Probe Thresholds</b>	

**Table 29: Real-Time Performance Monitoring Configuration Options** (*continued*)

Field	Description
Successive Lost Probes	Total number of probes that must be lost successively to trigger a probe failure and generate a system log message. The range is 0 to 15 probes.
Lost Probes	Total number of probes that must be lost to trigger a probe failure and generate a system log message. The range is 0 to 15 probes.
Round Trip Time	Total round-trip time (in microseconds) from the Services Router to the remote server, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Jitter	Total jitter (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Standard Deviation	Maximum allowable standard deviation (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Egress Time	Total one-way time (in microseconds) from the router to the remote server, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Ingress Time	Total one-way time (in microseconds) from the remote server to the router, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Jitter Egress Time	Total outbound-time jitter (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Jitter Ingress Time	Total inbound-time jitter (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Egress Standard Deviation	Maximum allowable standard deviation of outbound times (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.
Ingress Standard Deviation	Maximum allowable standard deviation of inbound times (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.

**Displaying Real-Time Performance Monitoring Information**

For each real-time performance monitoring test configured on the routing platform, monitoring information includes the round-trip time, jitter, and standard deviation. To view this information, select **Monitor > RPM** in the J-Web interface, or enter the `show services rpm` CLI command.

To display the results of the most recent real-time performance monitoring probes, enter the `show services rpm probe-results` CLI command:

```
user@host> show services rpm probe-results
Owner: p1, Test: t1
Target address: 10.8.4.1, Source address: 10.8.4.2, Probe type: icmp-ping
Destination interface name: lt-0/0/0.0
Test size: 10 probes
Probe results:
  Response received, Sun Jul 10 19:07:34 2005
  Rtt: 50302 usec
Results over current test:
  Probes sent: 2, Probes received: 1, Loss percentage: 50
  Measurement: Round trip time
    Minimum: 50302 usec, Maximum: 50302 usec, Average: 50302 usec,
    Jitter: 0 usec, Stddev: 0 usec
Results over all tests:
  Probes sent: 2, Probes received: 1, Loss percentage: 50
  Measurement: Round trip time
    Minimum: 50302 usec, Maximum: 50302 usec, Average: 50302 usec,
    Jitter: 0 usec, Stddev: 0 usec
```

## 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 30 on page 274.

**Table 30: Health Metrics**

<b>Metric:</b>	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
<b>Metric:</b>	Errors out
Description	Number of outbound packets that contained errors, preventing them from being transmitted
MIB name	IF-MIB (RFC 2233)
Variable name	ifOutErrors



**Table 30: Health Metrics** *(continued)*

Variable OID	.1.3.6.1.31.2.2.1.20
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	Logical interfaces
<b>Metric:</b>	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
<b>Metric:</b>	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
Allowable range	To be baselined
Managed objects	Logical interfaces
<b>Metric:</b>	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)

**Table 30: Health Metrics** *(continued)*

Managed objects	Logical interfaces
<b>Metric:</b>	Label Switched Path (LSP) state
Description	Operational state of an MPLS label-switched path
MIB name	MPLS-MIB
Variable name	<code>mplsLspState</code>
Variable OID	<code>mplsLspEntry.2</code>
Frequency (mins)	60
Allowable range	2 (up)
Managed objects	All label-switched paths in the network
<b>Metric:</b>	Component operating status
Description	Operational status of a router hardware component
MIB name	JUNIPER-MIB
Variable name	<code>jnxOperatingState</code>
Variable OID	<code>.1.3.6.1.4.1.2636.1.13.1.6</code>
Frequency (mins)	60
Allowable range	2 (running) or 3 (ready)
Managed objects	All components in each Juniper Networks router
<b>Metric:</b>	Component operating temperature
Description	Operational temperature of a hardware component, in Celsius
MIB name	JUNIPER-MIB
Variable name	<code>jnxOperatingTemp</code>
Variable OID	<code>.1.3.6.1.4.1.2636.1.13.1.7</code>
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All components in a chassis
<b>Metric:</b>	System up time
Description	Time, in milliseconds, that the system has been operational.
MIB name	MIB-2 (RFC 1213)

**Table 30: Health Metrics** *(continued)*

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
<b>Metric:</b>	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
<b>Metric:</b>	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
<b>Metric:</b>	SNMP community violations
Description	Number of valid SNMP communities used to attempt invalid operations (for example, attempting to perform SNMP <b>Set</b> requests)
MIB name	MIB-2 (RFC 1213)
Variable name	snmpInBadCommunityUses
Variable OID	snmp.5
Frequency (hours)	24

**Table 30: Health Metrics** *(continued)*

Allowable range	To be baselined
Managed objects	Each router
<b>Metric:</b>	Redundancy switchover
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
<b>Metric:</b>	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 6 for ready/online states. See jnxFruOfflineReason in the event of a FRU failure.
Managed objects	All FRUs in all Juniper Networks routers.
<b>Metric:</b>	Rate of tail-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 per interface in the provider network, when CoS is enabled.
<b>Metric:</b>	Interface utilization: octets received

**Table 30: Health Metrics** (*continued*)

Description	Total number of octets received on the interface, including framing characters.
MIB name	IF-MIB
Variable name	ifInOctets
Variable OID	.1.3.6.1.2.1.2.2.1.10.x
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All operational interfaces in the network
<b>Metric:</b>	Interface utilization: octets transmitted
Description	Total number of octets transmitted out of the interface, including framing characters.
MIB name	IF-MIB
Variable name	ifOutOctets
Variable OID	.1.3.6.1.2.1.2.2.1.16.x
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All operational interfaces in the network



**NOTE:** Byte counts vary depending on interface type, encapsulation used and PIC supported. For example, with vlan-ccc encapsulation on a 4xFE, GE, or GE 1Q PIC, the byte count includes framing and control word overhead. (See Table 31 on page 279.)

**Table 31: Counter Values for vlan-ccc Encapsulation**

PIC Type	Encapsulation	Input (Unit Level)	Output (Unit Level)	SNMP
4xFE	vlan-ccc	Frame (no frame check sequence [FCS])	Frame (including FCS and control word)	ifInOctets, ifOutOctets
GE	vlan-ccc	Frame (no FCS)	Frame (including FCS and control word)	ifInOctets, ifOutOctets

**Table 31: Counter Values for vlan-ccc Encapsulation (continued)**

PIC Type	Encapsulation	input (Unit Level)	Output (Unit Level)	SNMP
GE IQ	vlan-ccc	Frame (no FCS)	Frame (including FCS and control word)	ifInOctets, ifOutOctets

SNMP traps are also a good mechanism to use for health management. For more information, see “Standard SNMP Traps” on page 143 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.

## 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 32 on page 280).

**Table 32: Performance Metrics**

<b>Metric:</b>	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
<b>Metric:</b>	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

**Table 32: Performance Metrics** (continued)

<b>Metric:</b>	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
<b>Metric:</b>	Memory utilization
Description	Utilization of memory on the Routing Engine and FPC.
MIB name	JUNIPER-MIB (Juniper Networks 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
<b>Metric:</b>	CPU load
Description	Average utilization over the past minute of a CPU.
MIB name	JUNIPER-MIB (Juniper Networks 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
<b>Metric:</b>	LSP utilization
Description	Utilization of the MPLS label-switched path.
MIB name	MPLS-MIB
Variable name	mplsPathBandwidth / (mplsLspOctets * 8)

**Table 32: Performance Metrics** (*continued*)

Variable OID	mplsLspEntry.21 and mplsLspEntry.3
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All label-switched paths in the network
<b>Metric:</b>	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 per interface in the network, once CoS is enabled.

This section includes the following topics:

- Measuring Class of Service on page 282
- Inbound Firewall Filter Counters per Class on page 283
- Monitoring Output Bytes per Queue on page 284
- Dropped Traffic on page 285

## 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:

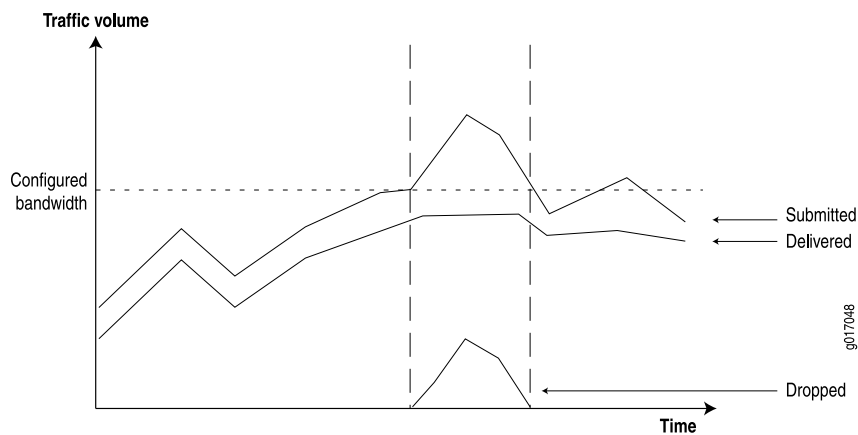
- Identify the type of packets that will be applied to this class. For example, include all customer traffic from a specific ingress edge interface within one class, or include all packets of a particular protocol such as voice over IP (VoIP).
- Identify the required deterministic behavior for each class. 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 7 on page 283.)

**Figure 7: 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 33 on page 284 shows additional filters used to match the other classes.

**Table 33: Inbound Traffic Per Class**

DSCP Value	Firewall Match Condition	Description
10	af1 1	Assured forwarding class 1 drop profile 1
12	af1 2	Assured forwarding class 1 drop profile 2
18	af2 1	Best effort class 2 drop profile 1
20	af2 2	Best effort class 2 drop profile 2
26	af3 1	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-specific Firewall Filter MIB presents the counter information in the variables shown in Table 34 on page 284.

**Table 34: 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 35 on page 285.)

**Table 35: 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-specific CoS MIB, which provides information shown in Table 36 on page 285

**Table 36: 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 or packets per interface per forwarding class
SNMP version	SNMPv2

## Dropped Traffic

You can calculate the amount of dropped traffic by subtracting the outbound traffic from the incoming traffic:

$$\text{Dropped} = \text{Inbound Counter} - \text{Outbound Counter}$$

You can also select counters from the CoS MIB, as shown in Table 37 on page 285.

**Table 37: Dropped Traffic Counters**

Indicator Name	Dropped Traffic
MIB	JUNIPER-COS-MIB

**Table 37: Dropped Traffic Counters** *(continued)*

Indicator Name	Dropped Traffic
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 7

# Juniper Networks Enterprise-Specific MIBs

- Interpreting the Structure of Management Information MIB on page 289
- Interpreting the Enterprise-Specific Chassis MIBs on page 295
- Interpreting the Enterprise-Specific Destination Class Usage MIB on page 387
- Interpreting the Enterprise-Specific BGP4 V2 MIB on page 389
- Interpreting the Enterprise-Specific Ping MIB on page 391
- Interpreting the Enterprise-Specific Traceroute MIB on page 405
- Interpreting the Enterprise-Specific RMON Events and Alarms MIB on page 407
- Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB on page 411
- Interpreting the Enterprise-Specific Source Class Usage MIB on page 413
- Interpreting the Enterprise-Specific Passive Monitoring MIB on page 415
- Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB on page 417
- Interpreting the Enterprise-Specific SONET APS MIB on page 419
- Interpreting the Enterprise-Specific IPSec Monitoring MIB on page 429
- Interpreting the Enterprise-Specific Ethernet MAC MIB on page 437
- Interpreting the Enterprise-Specific Interface MIB on page 439
- Interpreting the Enterprise-Specific VPN MIB on page 445
- Interpreting the Enterprise-Specific Flow Collection Services MIB on page 457
- Interpreting the Enterprise-Specific Services PIC MIB on page 461
- Interpreting the Enterprise-Specific Dynamic Flow Capture MIB on page 467
- Interpreting the Enterprise-Specific Chassis Forwarding MIB on page 475
- Interpreting the Enterprise-Specific System Log MIB on page 477
- Interpreting the Enterprise-Specific MPLS LDP MIB on page 481
- Interpreting the Enterprise-Specific Packet Forwarding Engine MIB on page 483
- Interpreting the Enterprise-Specific Event MIB on page 487
- Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB on page 489
- Interpreting the Enterprise-Specific Layer 2 Transport Protocol (L2TP) MIB on page 491

- Interpreting the Enterprise-Specific Real-Time Performance Monitoring (RPM) MIB on page 501
- Interpreting the Enterprise-Specific Class-of-Service MIB on page 509
- Interpreting the Enterprise-Specific IP Forward MIB on page 513
- Interpreting the Enterprise-Specific ATM Class-of-Service MIB on page 515
- Interpreting the Enterprise-Specific Firewall MIB on page 521
- Interpreting the Enterprise-Specific ATM MIB on page 523
- Interpreting the Enterprise-Specific Configuration Management MIB on page 533
- Interpreting the Enterprise-Specific IPv4 MIB on page 537
- Interpreting the Enterprise-Specific Alarm MIB on page 539
- Interpreting the Enterprise-Specific Resource Reservation Protocol (RSVP) MIB on page 541
- Interpreting the Enterprise-Specific MPLS MIB on page 543
- Interpreting the Enterprise-Specific Host Resources MIB on page 549
- Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB on page 551
- Interpreting the Enterprise-Specific MIMSTP MIB on page 553
- Interpreting the Enterprise-Specific L2ALD MIB on page 567
- Interpreting the Enterprise-Specific Utility MIB on page 569
- Interpreting the Enterprise-Specific AAA Objects MIB on page 573
- Interpreting the Enterprise-Specific Access Authentication Objects MIB on page 577
- Interpreting the Enterprise-Specific DNS Objects MIB on page 579
- Interpreting the Enterprise-Specific IPSec Generic Flow Monitoring Object MIB on page 581
- Interpreting the Enterprise-Specific IPSec VPN Objects MIB on page 595
- Interpreting the Enterprise-Specific Network Address Translation Objects MIB on page 599
- Interpreting the Enterprise-Specific Policy Objects MIB on page 603
- Interpreting the Enterprise-Specific Security Interface Extension Objects MIB on page 609
- Interpreting the VPN Certificate Objects MIB on page 613
- Interpreting the Enterprise-Specific Security Screening Objects MIB on page 615
- Interpreting the Enterprise-Specific LDP MIB on page 631
- Interpreting the Enterprise-Specific EX-Series SMI MIB on page 635
- Interpreting the Enterprise-Specific Analyzer MIB on page 637
- Interpreting the Enterprise-Specific VLAN MIB on page 641
- Interpreting the Enterprise-Specific Virtual Chassis MIB on page 645
- Interpreting the Enterprise-Specific PAE Extension MIB on page 647
- Interpreting the Enterprise-Specific Secure Access Port MIB on page 651

## Chapter 21

# 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/junos92/swconfig-net-mgmt/mib-jnx-smi.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-smi.txt).

The Structure of Management Information MIB space has five root branches:

- `jnxProducts` on page 289
- `jnxServices` on page 289
- `jnxMibs` on page 290
- `jnxTraps` on page 292
- `jnxExperiment` on page 292

### **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}`. This MIB file added the nodes to create the Juniper Networks security tree structure under the object node `jnxJsObjects`. In general, the prefix `jnxJs` is used to name the object identifiers and to designate them. This branch of the network describes the Juniper Networks services objects that provide enhanced network security. This MIB is currently supported only by JUNOS software with enhanced services.

The `jnxJsSecurity` node is designed to provide a branch for the security-related MIB definitions specific to the Juniper Networks security products. The next level object identifiers under `jnxJsSecurity` are:

- `jnxJsIf`—Whose object identifier is `{jnxJsSecurity 1}`.
- `jnxJsAuth`—Whose object identifier is `{jnxJsSecurity 2}`.
- `jnxJsCertificates`—Whose object identifier is `{jnxJsSecurity 3}`.
- `jnxJsPolicies`—Whose object identifier is `{jnxJsSecurity 4}`.
- `jnxJsIPSecVpn`—Whose object identifier is `{jnxJsSecurity 5}`.
- `jnxJsResources`—Whose object identifier is `{jnxJsSecurity 6}`.
- `jnxJsNAT`—Whose object identifier is `{jnxJsSecurity 7}`.
- `jnxJsScreening`—Whose object identifier is `{jnxJsSecurity 8}`.
- `jnxJsDhcp`—Whose object identifier is `{jnxJsSecurity 9}`.
- `jnxJsDnsRoot`—Whose object identifier is `{jnxJsSecurity 10}`.

The Juniper Networks enterprise-specific security MIBs include:

- AAA Objects MIB—Whose object identifier is `{jnxUserAAAMibRoot 1}`.
- Access Authentication Objects MIB—Whose object identifier is `{jnxJsAuth 1}`.
- DNS Objects MIB—Whose object identifier is `{jnxJsDns 1}`.
- IPSec Generic Flow Monitoring Objects MIB—Whose object identifier is `{jnxIpSecMibRoot 1}`.
- IPSec VPN Objects MIB—Whose object identifier is `{jnxJsIPSecVpn 1}`.
- Network Address Translation Objects MIB—Whose object identifier is `{jnxJsNAT 1}`.
- Policy Objects MIB—Whose object identifier is `{jnxJsPolicies 1}`.
- Security Interface Extension Objects MIB—Whose object identifier is `{jnxJsIf 1}`.
- VPN Certificate Objects MIB—Whose object identifier is `{jnxJsCertificates 1}`.
- Security Screening Objects MIB—Whose object identifier is `{jnxJsScreening 1}`.

For more information on these MIBs, see “Juniper Networks Enterprise-Specific MIBs” on page 123.

## **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}`.
- Firewall 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 Events and Alarms 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/SDH Interface Management MIB—Whose object identifier is {jnxMibs 20}.
- ATM class-of-service MIB—Whose object identifier is {jnxMibs 21}.
- IPsec Monitoring MIB—Whose object identifier is {jnxMibs 22}.
- Ethernet MAC MIB—Whose object identifier is {jnxMibs 23}.
- SONET APS MIB—Whose object identifier is {jnxMibs 24}.
- Chassis Definitions for Router Model 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}.
- Dynamic Flow Capture (DFC) MIB—Whose object identifier is {jnxMibs 33}.
- Chassis Forwarding MIB—Whose object identifier is {jnxMibs 34}.
- System Log MIB—Whose object identifier is {jnxMibs 35}.
- MPLS LDP MIB—Whose object identifier is {jnxMibs 36}.
- Event MIB—Whose object identifier is {jnxMibs 37}.
- IP Forward MIB—Whose object identifier is {jnxMibs 38}.
- Packet Forwarding Engine MIB—Whose object identifier is {jnxPfeMibRoot 1}.

- BFD MIB—Whose object identifier is {jnxBfdMibRoot 1}.
- Utility MIB—Whose object identifier is {jnxMibs 47}.
- L2ALD MIB—Whose object identifier is {jnxMibs 48}.
- L2TP MIB—Whose object identifier is {jnxMibs 49}.
- RPM MIB—Whose object identifier is {jnxMibs 50}.
- User AAA MIB—Whose object identifier is {jnxMibs 51}.

For more information on these MIBs, see “Juniper Networks Enterprise-Specific MIBs” on page 123.

## jnxTraps

---

The object identifier for the **jnxTraps** root branch of the Structure of Management Information MIB is {juniperMIB 4}. The **jnxTraps** root 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** root branch of the Structure of Management Information MIB is {juniperMIB 5}. The **jnxExperiment** root branch contains experimental Juniper Networks enterprise-specific MIBs. This is the top-level object identifier registry used by Juniper Networks 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 22

# 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 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 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 Definitions for Router Model MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt).

This chapter contains the following topics:

- `jnxBoxAnatomy` on page 295
- Chassis Traps on page 375
- Chassis Definitions for Router Model MIB on page 380
- MIB Objects for the M120 Router on page 381
- MIB Objects for the MX960 Ethernet Services Router on page 382
- MIB Objects for the MX480 Ethernet Services Router on page 382
- MIB Objects for the MX240 Ethernet Services Router on page 383
- MIB Objects for the EX-Series Ethernet Switches on page 383
- MIB Objects for the SRX 5600 Services Gateway on page 385
- MIB Objects for the SRX 5800 Services Gateway on page 385

## **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:

- Top-Level Objects on page 296
- `jnxContainersTable` on page 296

- `jnxContentsLastChange` on page 303
- `jnxContentsTable` on page 303
- `jnxLEDLastChange` on page 314
- `jnxLEDTable` on page 314
- `jnxFilledLastChange` on page 317
- `jnxFilledTable` on page 318
- `jnxOperatingTable` on page 327
- `jnxRedundancyTable` on page 335
- `jnxFruTable` on page 340
- `jnxBoxKernelMemoryUsedPercent` on page 375

## 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 303.

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 38 on page 298 through Table 45 on page 302 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 38 on page 298 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M40 router.

**Table 38: 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



**Table 38: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router (continued)**

Index	View	Level	Within	Type	Description	Count
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 39 on page 299 describes objects in the jnxContainersTable of an M20 router.

**Table 39: 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 40 on page 299 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M160 router.

**Table 40: 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

**Table 40: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router (continued)**

Index	View	Level	Within	Type	Description	Count
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 41 on page 300 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M10 router.

**Table 41: 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 42 on page 301 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M5 router.

**Table 42: 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 43 on page 301 describes objects contained in a jnxContainersEntry in the jnxContainersTable of a T640 routing node.

**Table 43: 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 44 on page 302 describes objects contained in a `jnxContainersEntry` in the `jnxContainersTable` of a T320 router.

**Table 44: `jnxContainersEntry` Objects in the `jnxContainersTable` of a T320 Router**

Index	View	Level	Within	Type	Description	Count
1	1	0	0	<code>jnxChassisT320.0</code>	Chassis frame	1
2	2	1	1	<code>jnxT320SlotPower.0</code>	PEM slot	2
4	3	1	1	<code>jnx320SlotFan.0</code>	Fan slot	3
7	1	1	1	<code>jnxT320SlotFPC.0</code>	FPC slot	8
8	1	2	7	<code>jnxT320MediaCardSpacePIC.0</code>	PIC slot	2
9	2	1	1	<code>jnxT320SlotHM.0</code>	Host slot	2
10	1	1	1	<code>jnxT320SlotFPB.0</code>	FPM slot	1
11	2	1	1	<code>jnxT320SlotSCG.0</code>	SCG slot	2
12	2	1	1	<code>jnxT320SlotCB.0</code>	CB slot	2
13	1	1	1	<code>jnxT320SlotCIP.0</code>	CIP slot	1
14	2	1	1	<code>jnxT320SlotSPMB.0</code>	SPMB slot	2
15	2	1	1	<code>jnxT320SlotSIB.0</code>	SIB slot	3

Table 45 on page 302 describes objects contained in a `jnxContainersEntry` in the `jnxContainersTable` of an M40e router.

**Table 45: `jnxContainersEntry` Objects in the `jnxContainersTable` of an M40e Router**

Index	View	Level	Within	Type	Description	Count
1	1	0	0	<code>jnxChassisM40e.0</code>	Chassis frame compartment	1
2	2	1	1	<code>jnxM40eSlotPower.0</code>	Power supply compartment	2
4	3	1	1	<code>jnxM40eSlotFan.0</code>	Fan compartment	4
6	2	1	1	<code>jnxM40eSlotSFM.0</code>	SFM slot	2
7	1	1	1	<code>jnxM40eSlotFPC.0</code>	FPC slot	8
8	1	2	7	<code>jnxM40eMediaCardSpacePIC.0</code>	PIC space	4
9	2	1	1	<code>jnxM40eSlotHM.0</code>	Host slot	2

**Table 45: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router (continued)**

Index	View	Level	Within	Type	Description	Count
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, whereas 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 46 on page 304 through Table 48 on page 311 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 Index—**jnxContentsL1Index**
- L2 Index—**jnxContentsL2Index**
- L3 Index—**jnxContentsL3Index**
- Type—**jnxContentsType**
- Description—**jnxContentsDescr**
- Serial Number—**jnxContentsSerialNo**
- Revision—**jnxContentsRevision**
- Installed—**jnxContentsInstalled**
- Part Number—**jnxContentsPartNo**

Table 46 on page 304 provides an example of **jnxContentEntry** objects in the **jnxContentTable** of an M20 router.

**Table 46: 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	–

**Table 46: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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

**Table 46: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	3	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX@ 2/0/*	AH4147	REV 07	0:0:00:00.00	750-002303
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 47 on page 307 provides an example of `jnxContentEntry` objects in the `jnxContentTable` of a T640 routing node.

**Table 47: `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	<code>jnxMidplaneT640.0</code>	Midplane	AX5633	REV 04	0:0:00:00.00	710-002726
2	2	0	0	<code>jnxT640Power.0</code>	PEM 1	MD21815	RevX02	0:0:00:00.00	740-002595
4	1	1	0	<code>jnxT640Fan.0</code>	Top left front fan	–	–	0:0:00:00.00	–
4	1	2	0	<code>jnxT640Fan.0</code>	Top left middle fan	–	–	0:0:00:00.00	–
4	1	3	0	<code>jnxT640Fan.0</code>	Top left rear fan	–	–	0:0:00:00.00	–
4	1	4	0	<code>jnxT640Fan.0</code>	Top right front fan	–	–	0:0:00:00.00	–
4	1	5	0	<code>jnxT640Fan.0</code>	Top right middle fan	–	–	0:0:00:00.00	–
4	1	6	0	<code>jnxT640Fan.0</code>	Top right rear fan	–	–	0:0:00:00.00	–
4	2	1	0	<code>jnxT640Fan.0</code>	Bottom left front fan	–	–	0:0:00:00.00	–
4	2	2	0	<code>jnxT640Fan.0</code>	Bottom left middle fan	–	–	0:0:00:00.00	–
4	2	3	0	<code>jnxT640Fan.0</code>	Bottom left rear fan	–	–	0:0:00:00.00	–
4	2	4	0	<code>jnxT640Fan.0</code>	Bottom right front fan	–	–	0:0:00:00.00	–

**Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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
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

**Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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
9	1	0	0	jnxT640HM.0	Host 0	–	–	0:19:19:30.95	–
9	2	0	0	jnxT640HM.0	Host 1	2108 6570 0292	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

**Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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 48 on page 311 provides an example of `jnxContentEntry` objects in the `jnxContentTable` of a T320 router.

**Table 48: `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	<code>jnxMidplaneT320.0</code>	Midplane	AY4527	Rev 01	(0) 0:00:00.00	710-004339
2	1	0	0	<code>jnxT320Power.0</code>	PEM 0	ML14099	Rev 01	(0) 0:00:00.00	–
4	1	1	0	<code>jnxT320Fan.0</code>	Top left front fan	–	–	(0) 0:00:00.00	–
4	1	2	0	<code>jnxT320Fan.0</code>	Top left middle fan	–	–	(0) 0:00:00.00	–
4	1	3	0	<code>jnxT320Fan.0</code>	Top left rear fan	–	–	(0) 0:00:00.00	–
4	1	4	0	<code>jnxT320Fan.0</code>	Top right front fan	–	–	(0) 0:00:00.00	–
4	1	5	0	<code>jnxT320Fan.0</code>	Top right middle fan	–	–	(0) 0:00:00.00	–
4	1	6	0	<code>jnxT320Fan.0</code>	Top right rear fan	–	–	(0) 0:00:00.00	–
4	2	1	0	<code>jnxT320Fan.0</code>	Bottom left front fan	–	–	(0) 0:00:00.00	–
4	2	2	0	<code>jnxT320Fan.0</code>	Bottom left middle fan	–	–	(0) 0:00:00.00	–
4	2	3	0	<code>jnxT320Fan.0</code>	Bottom left rear fan	–	–	(0) 0:00:00.00	–

**Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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
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	2108 6570 0286	REV 01	(32762924) 3 days, 19:00:29.24	740-005022
9	2	0	0	jnxT320HM.0	Host 1	2109 2900 0186	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

**Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router**  
(continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
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.
- **jnxLEDOriinator**—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 49 on page 316 through Table 51 on page 317 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**
- Associate index—**jnxLEDAssociateIndex**
- L1 Index—**jnxLEDL1Index**
- L2 Index—**jnxLEDL2Index**
- L3 Index—**jnxLEDL3Index**
- Originator—**jnxLEDOriinator**

- Description—jnxLEDDescr
- State—jnxLEDState

Table 49 on page 316 provides an example of jnxLEDEntry objects in the jnxLEDTable of an M20 router.

**Table 49: 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 50 on page 316 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T640 routing node.

**Table 50: 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

**Table 50: jnxLEDEntry Objects in the jnxLEDTable of a T640 Routing Node (continued)**

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
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 51 on page 317 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T320 router.

**Table 51: 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 52 on page 318 through Table 54 on page 324 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 52 on page 318 provides an example of `jnxFilledEntry` objects in the `jnxFilledTable` of an M20 router.

**Table 52: `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

**Table 52: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router** (continued)

Container Index	L1	L2	L3	Description	State
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
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

**Table 52: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router** (continued)

Container Index	L1	L2	L3	Description	State
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 53 on page 321 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T640 routing node.

**Table 53: 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

**Table 53: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node**  
(continued)

Container Index	L1	L2	L3	Description	State
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
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



**Table 53: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node**  
(continued)

Container Index	L1	L2	L3	Description	State
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
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

**Table 53: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node**  
(continued)

Container Index	L1	L2	L3	Description	State
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 54 on page 324 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T320 router.

**Table 54: 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

**Table 54: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router** *(continued)*

Container Index	L1	L2	L3	Description	State
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
7	6	1	0	FPC slot 5	Empty

**Table 54: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router** *(continued)*

Container Index	L1	L2	L3	Description	State
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

**Table 54: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router** *(continued)*

Container Index	L1	L2	L3	Description	State
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 `jnxOperatingTable` 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 55 on page 330 through Table 57 on page 333 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 55 on page 330 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of an M20 router.

**Table 55: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router**

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	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	671088	6	0	67038195	0:0:00:35.41	64
7	1	0	0	FPC @ 0/*/*	Running	31	0	0	83886	3	0	67035034	0:0:01:06.91	8
7	2	0	0	FPC @ 1/*/*	Running	33	0	0	83886	4	0	67034422	0:0:01:13.04	8
7	3	0	0	FPC @ 2/*/*	Running	31	0	0	83886	3	0	67033809	0:0:01:19.18	8
9	1	0	0	Routing Engine 0	Running	29	4	0	802738	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 56 on page 330 provides an example of `jnxOperatingEntry` objects in the `jnxOperatingTable` of a T640 routing node.

**Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node**

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	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	–	–	–	–	–	–	–	–



**Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node** (continued)

Index	L1	L2	B	Description	State	Temp	CPU	BR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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

**Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node** (continued)

Index	L1	L2	B	Description	State	Temp	CPU	BR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	2	1	0	FPC @ 1/0/* top temp. sensor	Running	35	0	0	0	0	0	0	0:18:56:48.81	0
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

**Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node** (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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
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 57 on page 333 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of a T320 router.

**Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router**

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	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

**Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router**  
(continued)

Index	L1	L2	L3	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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	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

**Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router**  
(continued)

Index	L1	L2	L3	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
11	1	0	0	SCG 0	Running	33	0	0	0	0	0	0	(0) 0:00:00.00	0
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 58 on page 337 through Table 60 on page 339 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 58 on page 337 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of an M20 router.

**Table 58: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router**

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	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 59 on page 338 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of a T640 routing node.

**Table 59: `jnxRedundancyEntry` Objects in the `jnxRedundancyTable` of a T640 Routing Node**

Index	I1	I2	I3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	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 60 on page 339 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of a T320 router.

**Table 60: `jnxRedundancyEntry` Objects in the `jnxRedundancyTable` of a T320 Router**

Index	IL	I2	B	Description	Config	State	Count	Time	Reason	Heart beat	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

**Table 60: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router**  
(continued)

Index	IL	I2	B	Description	Config	State	Count	Time	Reason	Heart beat	Timeout	Elapsed	Loss
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 61 on page 343 through Table 66 on page 366 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 61 on page 343 provides an example of `jnxFruContent` objects in the `jnxFruTable` for an M10 router.

**Table 61: `jnxFruContents` Objects in the `jnxFruTable` of an M10 Router**

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	264319
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

**Table 61: jnxFruContents Objects in the jnxFruTable of an M10 Router** *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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          REV 03   710-001950   HB1590        M10
Midplane         Rev 03   740-002498   LK33505       DC
Power Supply A

```

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 62 on page 346 provides an example of `jnxFruContent` objects in the `jnxFruTable` for an M20 router.

**Table 62: JnxFruContents Objects in the jnxFruTable of an M20 Router**

Index	I1	I2	I3	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 1	controlBoard	2	Online	29	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexible PicConcentrator	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexible PicConcentrator-	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexible PicConcentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexible PicConcentrator-	4	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
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



**Table 62: JnxFruContents Objects in the jnxFruTable of an M20 Router (continued)**

Index	IL	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
```

Slot	State	Temp (C)	CPU Utilization (%)		Memory DRAM (MB)	Utilization (%)	
			Total	Interrupt		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 63 on page 349 provides an example of `jnxFruContent` objects in the `jnxFruTable` for an M160 router.

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router**

Index	IL	P	B	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

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router** *(continued)*

Index	1L	2	B	Name	Type	Skt	State	Temp	Offline	PowerOff	PowerOn	Uptime
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	switchingAndFor wardingMode	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	Configured to power off	0:00:03.22	0:00:00.00	0
6	4	2	0	SFM 3 SPR	switchingAnd ForwardingMode	4	Offline	0	Configured to power off	0:00:03.22	0:00:00.00	0

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)**

Index	1	2	B	Name	Type	Slt	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	1	0	0	FPC @ 0/*/*	flexiblePic Concentrator	1	Offline	0	Configured to power off	0:00:02.28	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePic Concentrator	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/*	portInterface Card	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterface Card	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

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router** *(continued)*

Index	1L	2	B	Name	Type	Skt	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	Offline	0	Config- ured to power off	0:00:02.28	0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	Offline	0	Config- ured to power off	0:00:02.28	0:00:00.00	0
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

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)**

Index	IL	I2	B	Name	Type	Skt	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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/*	portInterfaceCard	8	Offline	0	Config- ured to power off	0:00:02.37	0:00:00.00	0
jnxFruName												
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

**Table 63: jnxFruContents Objects in the jnxFruTable of an M160 Router** (continued)

Index	L1	L2	L3	Name	Type	Skt	State	Temp	Offline	PowerOff	PowerOn	Uptime
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	craftInterfacePanel-	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   AARCH00        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 0C192
  CPU             REV 05    710-001217   AF5950
  PIC 0           REV 01    750-003063   HB2029        1x 0C-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 0C-12 SONET, SMIR
  PIC 1           REV 02    750-E3-PIC   AC1903        4x E3
  PIC 2           REV 02    750-001020   AA8944        1x 0C-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 64 on page 356 provides an example of `jnxFruContent` objects in the `jnxFruTable` for an M40 router.

**Table 64: jnxFruContents Objects in the jnxFruTable of an M40 Router**

Index	I1	I2	I3	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/*/*	flexiblePic Concentrator	1	Online	28	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexible PicConcentrator	2	Online	29	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexible PicConcentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePic Concentrator	4	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePic Concentrator	5	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePic Concentrator	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/*/*	flexiblePic Concentrator	7	Online	28	None	0:0:00:00.00	0:0:00:00.00	0

**Table 64: jnxFruContents Objects in the jnxFruTable of an M40 Router (continued)**

Index	1	2	3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	8	0	0	FPC @ 7/*/*	flexible PicConcentrator	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
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

**Table 64: jnxFruContents Objects in the jnxFruTable of an M40 Router (continued)**

Index	1L	2	13	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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-NxDS0 @ 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
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

**Table 64: jnxFruContents Objects in the jnxFruTable of an M40 Router (continued)**

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
SCB               REV 07   710-000075   AA1033         RE-1.0
FPC 0             REV 01   710-001292   AB8159         Internet Processor I
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 (C)	Utilization (%)	DRAM (MB)	Heap	Buffer
Slot	State	Total	Interrupt			
0	Online	28	2	0	8	11
1	Online	29	7	0	8	21
2	Empty	0	0	0	0	0
3	Online	24	17	0	8	22
4	Online	27	1	0	8	6
5	Empty	0	0	0	0	0
6	Online	28	1	0	8	7
7	Empty	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 65 on page 361 provides an example of jnxFruContent objects in the jnxFruTable for an M40e router.

**Table 65: JnxFruContents Objects in the jnxFruTable of an M40e Router**

Index	IL	I2	B	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	switchingAnd ForwardingModule-	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
6	1	2	0	SFM 0 SPR	switchingAnd ForwardingModule	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAnd ForwardingModule-	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	switchingAnd ForwardingModule-	2	Online	42	None	0:0:00:21.95	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePic Concentrator	1	Online	41	None	0:0:00:21.85	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexible PicConcentrator	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexible PicConcentrator	3	Online	43	None	0:0:00:21.87	0:0:00:00.00	0

**Table 65: JnxFruContents Objects in the jnxFruTable of an M40e Router** *(continued)*

Index	IL	I2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	4	0	0	FPC @ 3/*/*	flexible PicConcentrator	4	Online	38	None	0:0:00:21.89	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePic Concentrator	5	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePic Concentrator	6	Online	46	None	0:0:00:21.91	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePic Concentrator	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePic Concentrator	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



**Table 65: JnxFruContents Objects in the jnxFruTable of an M40e Router (continued)**

Index	IL	I2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

**Table 65: JnxFruContents Objects in the jnxFruTable of an M40e Router** (continued)

Index	IL	I2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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 66 on page 366 provides an example of `jnxFruContent` objects in the `jnxFruTable` for a T640 routing node.

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**

Index	1	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	power EntryModule	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	power EntryModule-	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

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	1	2	3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	1	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	IL	I	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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/*/*	<del>flexiblePic Concentrator</del>	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	1	0	FPC @ 4/0/* top temp. sensor	<del>flexiblePic Concentrator</del>	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	<del>flexiblePic Concentrator</del>	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	<del>flexiblePic Concentrator</del>	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	<del>flexiblePic Concentrator</del>	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	<del>flexiblePic Concentrator</del>	6	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	<del>flexiblePic Concentrator</del>	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	<del>flexiblePic Concentrator</del>	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	<del>flexiblePic Concentrator</del>	7	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	<del>flexiblePic Concentrator</del>	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	1	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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	<del>flexiblePic Concentrator</del>	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	<del>portInterfaceCard</del>	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	<del>portInterfaceCard</del>	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	<del>portInterfaceCard</del>	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	<del>portInterfaceCard</del>	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/*	<del>portInterfaceCard</del>	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/*	<del>portInterface- Card</del>	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/*	<del>portInterface - Card</del>	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/*	<del>portInterface- Card</del>	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	<del>portInterface- Card</del>	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	<del>portInterface- Card</del>	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0



**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	1	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	1	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterfaceCard	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterfaceCard	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	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
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	clockGenerator	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	Unknown	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

**Table 66: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node**  
(continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
```

### ***jnxBoxKernelMemoryUsedPercent***

The object identifier for `jnxBoxKernelMemoryUsedPercent` is `jnxBoxAnatomy 16`. This object represents the amount of the kernel memory used, expressed as a percentage of the total available memory. The object shows 0 if the kernel memory usage is unavailable or inapplicable. When the kernel memory usage exceeds 80 percent, a system log message is logged and an RMON rising threshold trap is generated if RMON health monitoring is enabled for the device.

## **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 131.

These traps are defined as follows:

- **Power failure (`jnxPowerSupplyFailure`)**—Sent when the power supply, router circuit breaker, or power circuit fails, or when there is a power outage. When only one of the power supplies has failed, 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. This trap is repeated every hour until the power supply is restored.
- **Fan failure (`jnxFanFailure`)**—Sent when the fan fuse blows or when the fan wiring shorts out. When 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. This trap is repeated every hour until the fan failure is fixed. 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`)**—Sent 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. This trap is repeated every minute until the temperature is brought down to normal. 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 (jnxFruPowerOff)**—Sent when the specified FRU has been powered off in the chassis.

The **jnxFruPowerOff** trap is also sent in the following scenarios:

- When an FRU that is controlled using inter-process communication (IPC) goes offline or is removed from the chassis. For example, a switch interface board (SIB).
- When an FRU that does not have a backup unit goes offline or is removed from the chassis. For example, a Flexible PIC Concentrator (FPC).



**NOTE:** When a SONET Clock Generator (SCG) is taken offline, the unit is not powered down. Therefore, **jnxFruPowerOff** or **jnxFruPowerOn** traps are not sent when the unit is taken online or offline.

---

- **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. However, when an FRU that does not have a backup unit goes offline, JUNOS software generates the **jnxFruPowerOff** trap instead of the **jnxFruOffline** trap. Typically, a **jnxFruOffline** trap is generated to inform the backup FRU about the status of the primary FRU so that the backup FRU can take over when the primary FRU goes offline.

The following are some scenarios when `jnxFruOffline` traps are generated:

- When a PFE Clock Generator (PCG) goes offline (M40e)
- When a Sonnet Clock Generator goes offline (T series)
- When a Line Card Chassis goes offline (TX4 internet routing node)
- When a Routing Engine 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. On M120 and M320 routers, this trap is sent if the revision number for the ATM2 PIC FPGA is less than 8B44(4).

For more information on Chassis MIB traps, see “Standard SNMP Traps” on page 143 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.

This section contains the following topics:

- SNMPv1 Trap Format on page 377
- SNMPv2 Trap Format on page 378

## SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in Table 67 on page 377. To view the SNMPv1 chassis-related traps, see “Standard SNMP Traps” on page 143 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 131.

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 67: SNMP Version 1 Trap Format**

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
<code>jnxFanFailure</code>	1.3.6.1.4.1.2636.4.1	6	2
<code>jnxFanOK</code>	1.3.6.1.4.1.2636.4.2	6	2
<code>jnxFruCheck</code>	1.3.6.1.4.1.2636.4.1	6	12

**Table 67: SNMP Version 1 Trap Format** (continued)

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
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 68 on page 378.

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 68: SNMP Version 2 Trap Format**

Trap Name	snmpTrapOID	Description
jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	The fan fuse has blown or the fan wiring has shorted out. This trap is generated every hour until the fan failure is fixed.



**Table 68: SNMP Version 2 Trap Format** (continued)

Trap Name	snmpTrapOID	Description
jnxFanOK	1.3.6.1.4.1.2636.4.2.2	The fan has recovered from a 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 diagnostic state. The revision number for the ATM2 PIC FPGA on an M120 or M320 router is less than 8B44(4).
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 have failed or the room temperature has increased significantly. This trap is repeated every minute until the temperature is brought down to normal.
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	The power supply, router circuit breaker, or power circuit failed, or there has been a power outage. This trap is generated every hour until the power supply is restored.
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	The power supply has recovered from a failure.
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 an overtemperature condition.

## Chassis Definitions for Router Model MIB

The enterprise-specific Chassis Definitions for Router Model MIB contain 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 Router Model MIB provide information that changes less often.

The last number in each `sysObjectID`, shown in Table 69 on page 380, corresponds to the router model and therefore does not change.

**Table 69: 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
M120	1.3.6.1.4.1.2636.1.1.1.2.18	jnxProductNameM120
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
MX960	1.3.6.1.4.1.2636.1.1.1.2.21	jnxProductNameX960
SRX 5600	1.3.6.1.4.1.2636.1.1.1.2.28	jnxProductNameSRX5600
SRX 5800	1.3.6.1.4.1.2636.1.1.1.2.26	jnxProductNameSRX5800
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 Router Model MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chas-defines.txt).

## MIB Objects for the M120 Router

New Chassis Management Information Base (MIB) objects for the M120 router include:

```

jnxProductLineM120      OBJECT IDENTIFIER ::= { jnxProductLine      18 }
jnxProductNameM120      OBJECT IDENTIFIER ::= { jnxProductName      18 }
jnxProductModelM120     OBJECT IDENTIFIER ::= { jnxProductModel     18 }
jnxProductVariationM120 OBJECT IDENTIFIER ::= { jnxProductVariation 18 }
jnxChassisM120          OBJECT IDENTIFIER ::= { jnxChassis          18 }
jnxSlotM120             OBJECT IDENTIFIER ::= { jnxSlot             18 }
    jnxM120SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotM120 1 }
    jnxM120SlotFEB       OBJECT IDENTIFIER ::= { jnxSlotM120 2 }
    jnxM120SlotHM        OBJECT IDENTIFIER ::= { jnxSlotM120 3 }
    jnxM120SlotPower     OBJECT IDENTIFIER ::= { jnxSlotM120 4 }
    jnxM120SlotFan       OBJECT IDENTIFIER ::= { jnxSlotM120 5 }
    jnxM120SlotCB        OBJECT IDENTIFIER ::= { jnxSlotM120 6 }
    jnxM120SlotFPB       OBJECT IDENTIFIER ::= { jnxSlotM120 7 }

jnxMediaCardSpaceM120   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   18 }
    jnxM120MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceM120 1 }

jnxMidplaneM120         OBJECT IDENTIFIER ::= { jnxBackplane        18 }
jnxModuleM120           OBJECT IDENTIFIER ::= { jnxModule           18 }
    jnxM120FEB           OBJECT IDENTIFIER ::= { jnxModuleM120     1 }

```



**NOTE:** The M120 router does not support the enterprise-specific Dynamic Flow Capture MIB.

Sample command output from the `show chassis hardware` command for the M120 router is listed below.

```
user@host> show chassis hardware
```

Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			JN000019AC	M120
Midplane	REV 01	710-011382	RB3003	M120 Midplane
FPM Board	REV 01	710-011407	CK9165	M120 FPM Board
FPM Display	REV 01	710-011405	CE0032	M120 FPM Display
FPM CIP	REV 01	710-011410	CE0058	M120 FPM CIP
PEM 1	Rev 01	740-011935	RG10165	DC Power Entry Module
Routing Engine 0	REV 00	740-014082	1000604605	RE-A-2000
Routing Engine 1	REV 00	740-014082	1000604601	RE-A-2000
CB 0	REV 03	710-011403	CM8335	M120 Control Board
CB 1	REV 03	710-011403	CM8340	M120 Control Board
FPC 0	REV 01	710-012879	CH1622	M120 CFPC OC192
PIC 0		BUILTIN	BUILTIN	1x OC-192 SONET XFP
Xcvr 0		NON-JNPR	T05J32698	XFP-OC192-SR
FPC 1	REV 01	710-012882	CE0062	M120 CFPC 10GE
PIC 0		BUILTIN	BUILTIN	1x 10GE(LAN/WAN) XFP
Xcvr 0		NON-JNPR	T05A02227	XFP-10G-ER
FPC 2	REV 01	710-011388	CJ9092	M120 FPC Type 1
PIC 0	REV 16	750-007444	HS1526	1x CHOC3 IQ SONET, SMIR
PIC 1	REV 12	750-005637	HT0533	4x CHDS3 IQ
PIC 2	REV 15	750-005634	HN1903	1x CHOC12 IQ SONET, SMIR

PIC 3	REV 15	750-007631	NB5006	10x CHE1 IQ
Board B	REV 01	710-011390	CJ9109	M120 FPC Mezz Board
FPC 3	REV 03	710-011393	CJ9231	M120 FPC Type 2
PIC 0	REV 05	750-010472	JE3146	1x OC-48 ATM-II IQ
Xcvr 0	REV 01	740-009028	P5F05WU	SFP-SR
PIC 1	REV 13	750-001901	HB4231	4x OC-12 SONET, SMIR
PIC 2	REV 15	750-008155	HX5442	2x G/E IQ, 1000 BASE
Xcvr 0	REV	740-007326	P11E5RR	SFP-SX
Xcvr 1	REV 01	740-009029	4C81050	UNKNOWN
PIC 3	REV 16	750-008155	HZ8871	2x G/E IQ, 1000 BASE
Xcvr 0	REV 01	740-011613	P8E2KGF	SFP-SX
Xcvr 1	REV 01	740-011782	P6M1E5X	SFP-SX
Board B	REV 02	710-011395	CN3750	M120 FPC Mezz
FPC 4	REV 01	710-011388	CJ9089	M120 FPC Type 1
PIC 0	REV 03	750-002911	AJ2279	4x F/E, 100 BASE-TX
PIC 1	REV 15	750-005634	HN0435	1x CHOC12 IQ SONET, SMIR
PIC 2	REV 02	750-003064	HD4548	4x T1, RJ48
PIC 3	REV 04	750-011209	JC8254	Adaptive Services-II
Board B	REV 01	710-011390	CJ9111	M120 FPC Mezz Board
FPC 5	REV 01	710-011388	CJ9360	M120 FPC Type 1
PIC 0	REV 08	750-007631	HK0212	10x CHE1 IQ
PIC 1	REV 05	750-003034	BD8705	4x OC-3 SONET, SMIR
PIC 2	REV 11	750-007643	NA5967	1x G/E IQ, 1000 BASE
Xcvr 0	REV 01	740-007326	P4R0PNZ	SFP-SX
PIC 3	REV 16	750-007444	HS1501	1x CHSTM1 IQ SDH, SMIR
Board B	REV 01	710-011390	CJ9099	M120 FPC Mezz Board
FEB 0	REV 04	710-011663	CJ9364	M120 FEB
FEB 1	REV 04	710-011663	CJ9385	M120 FEB
FEB 2	REV 02	710-015795	CP6830	M120 FEB
FEB 3	REV 01	710-011663	CM2585	M120 FEB
FEB 4	REV 04	710-011663	CJ9416	M120 FEB
FEB 5	REV 01	710-011663	CM2600	M120 FEB

## MIB Objects for the MX960 Ethernet Services Router

The Chassis MIB objects for the MX960 Ethernet Services Router include:

jnxProductLineX960	OBJECT IDENTIFIER ::= { jnxProductLine 21 }
jnxProductNameX960	OBJECT IDENTIFIER ::= { jnxProductName 21 }
jnxProductModelX960	OBJECT IDENTIFIER ::= { jnxProductModel 21 }
jnxProductVariationX960	OBJECT IDENTIFIER ::= { jnxProductVariation 21 }
jnxChassisX960	OBJECT IDENTIFIER ::= { jnxChassis 21 }
jnxSlotX960	OBJECT IDENTIFIER ::= { jnxSlot 21 }
jnxX960SlotFPC	OBJECT IDENTIFIER ::= { jnxSlotX960 1 }
jnxX960SlotHM	OBJECT IDENTIFIER ::= { jnxSlotX960 2 }
jnxX960SlotPower	OBJECT IDENTIFIER ::= { jnxSlotX960 3 }
jnxX960SlotFan	OBJECT IDENTIFIER ::= { jnxSlotX960 4 }
jnxX960SlotCB	OBJECT IDENTIFIER ::= { jnxSlotX960 5 }
jnxX960SlotFPB	OBJECT IDENTIFIER ::= { jnxSlotX960 6 }
jnxMediaCardSpaceX960	OBJECT IDENTIFIER ::= { jnxMediaCardSpace 21 }
jnxX960MediaCardSpacePIC	OBJECT IDENTIFIER ::= { jnxMediaCardSpaceX960 1 }
jnxMidplaneX960	OBJECT IDENTIFIER ::= { jnxBackplane 21 }

## MIB Objects for the MX480 Ethernet Services Router

The Chassis MIB objects for the MX480 Ethernet Services Router include:

```

jnxProductLineMX480      OBJECT IDENTIFIER ::= { jnxProductLine      25 }
jnxProductNameMX480      OBJECT IDENTIFIER ::= { jnxProductName      25 }
jnxProductModelMX480     OBJECT IDENTIFIER ::= { jnxProductModel     25 }
jnxProductVariationMX480 OBJECT IDENTIFIER ::= { jnxProductVariation 25 }
jnxChassisMX480          OBJECT IDENTIFIER ::= { jnxChassis          25 }

jnxSlotMX480             OBJECT IDENTIFIER ::= { jnxSlot             25 }
jnxMX480SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotMX480 1 }
jnxMX480SlotHM           OBJECT IDENTIFIER ::= { jnxSlotMX480 2 }
jnxMX480SlotPower        OBJECT IDENTIFIER ::= { jnxSlotMX480 3 }
jnxMX480SlotFan           OBJECT IDENTIFIER ::= { jnxSlotMX480 4 }
jnxMX480SlotCB           OBJECT IDENTIFIER ::= { jnxSlotMX480 5 }
jnxMX480SlotFPB          OBJECT IDENTIFIER ::= { jnxSlotMX480 6 }

jnxMediaCardSpaceMX480   OBJECT IDENTIFIER ::= { jnxMediaCardSpace  25 }
jnxMX480MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX480 1 }

jnxMidplaneMX480         OBJECT IDENTIFIER ::= { jnxBackplane         25 }

```

## MIB Objects for the MX240 Ethernet Services Router

---

The Chassis MIB objects for the MX240 Ethernet Services Router include:

```

jnxProductLineMX240      OBJECT IDENTIFIER ::= { jnxProductLine      29 }
jnxProductNameMX240      OBJECT IDENTIFIER ::= { jnxProductName      29 }
jnxProductModelMX240     OBJECT IDENTIFIER ::= { jnxProductModel     29 }
jnxProductVariationMX240 OBJECT IDENTIFIER ::= { jnxProductVariation 29 }
jnxChassisMX240          OBJECT IDENTIFIER ::= { jnxChassis          29 }

jnxSlotMX240             OBJECT IDENTIFIER ::= { jnxSlot             29 }
jnxMX240SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotMX240 1 }
jnxMX240SlotHM           OBJECT IDENTIFIER ::= { jnxSlotMX240 2 }
jnxMX240SlotPower        OBJECT IDENTIFIER ::= { jnxSlotMX240 3 }
jnxMX240SlotFan           OBJECT IDENTIFIER ::= { jnxSlotMX240 4 }
jnxMX240SlotCB           OBJECT IDENTIFIER ::= { jnxSlotMX240 5 }
jnxMX240SlotFPB          OBJECT IDENTIFIER ::= { jnxSlotMX240 6 }

jnxMediaCardSpaceMX240   OBJECT IDENTIFIER ::= { jnxMediaCardSpace  29 }
jnxMX240MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX240 1 }

jnxMidplaneMX240         OBJECT IDENTIFIER ::= { jnxBackplane         29 }

```

## MIB Objects for the EX-Series Ethernet Switches

---

The Chassis MIB objects for the EX-series Ethernet switches include:

```

jnxProductLineEX3200     OBJECT IDENTIFIER ::= { jnxProductLine      30 }
jnxProductNameEX3200     OBJECT IDENTIFIER ::= { jnxProductName      30 }
jnxProductModelEX3200    OBJECT IDENTIFIER ::= { jnxProductModel     30 }
jnxProductVariationEX3200 OBJECT IDENTIFIER ::= { jnxProductVariation 30 }
jnxProductEX3200port24T  OBJECT IDENTIFIER ::= { jnxProductVariationEX3200 1 }

jnxProductEX3200port24P  OBJECT IDENTIFIER ::= { jnxProductVariationEX3200 2 }

```

```

jnxProductEX3200port48T OBJECT IDENTIFIER ::= { jnxProductVariationEX3200 3 }
jnxProductEX3200port48P OBJECT IDENTIFIER ::= { jnxProductVariationEX3200 4 }

jnxChassisEX3200          OBJECT IDENTIFIER ::= { jnxChassis          30 }

jnxSlotEX3200             OBJECT IDENTIFIER ::= { jnxSlot             30 }
  jnxEX3200SlotFPC        OBJECT IDENTIFIER ::= { jnxEX3200SlotFPC 1 }
    jnxEX3200SlotPower    OBJECT IDENTIFIER ::= { jnxEX3200SlotFPC 1 }
    jnxEX3200SlotFan      OBJECT IDENTIFIER ::= { jnxEX3200SlotFPC 2 }
    jnxEX3200SlotRE       OBJECT IDENTIFIER ::= { jnxEX3200SlotFPC 3 }

jnxMediaCardSpaceEX3200   OBJECT IDENTIFIER ::= { jnxMediaCardSpace 30 }
  jnxEX3200MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceEX3200 1 }
}

jnxModuleEX3200           OBJECT IDENTIFIER ::= { jnxModule          30 }
  jnxEX3200FPC             OBJECT IDENTIFIER ::= { jnxModuleEX3200 1 }
    jnxEX3200Power         OBJECT IDENTIFIER ::= { jnxEX3200FPC 1 }
    jnxEX3200Fan           OBJECT IDENTIFIER ::= { jnxEX3200FPC 2 }
    jnxEX3200RE            OBJECT IDENTIFIER ::= { jnxEX3200FPC 3 }

jnxProductLineEX4200      OBJECT IDENTIFIER ::= { jnxProductLine     31 }
jnxProductNameEX4200      OBJECT IDENTIFIER ::= { jnxProductName     31 }
jnxProductModelEX4200     OBJECT IDENTIFIER ::= { jnxProductModel    31 }
jnxProductVariationEX4200 OBJECT IDENTIFIER ::= { jnxProductVariation 31 }
  jnxProductEX4200port24T OBJECT IDENTIFIER ::= { jnxProductVariationEX4200 1 }

  jnxProductEX4200port24P OBJECT IDENTIFIER ::= { jnxProductVariationEX4200 2 }

  jnxProductEX4200port48T OBJECT IDENTIFIER ::= { jnxProductVariationEX4200 3 }

  jnxProductEX4200port48P OBJECT IDENTIFIER ::= { jnxProductVariationEX4200 4 }

  jnxProductEX4200port24F OBJECT IDENTIFIER ::= { jnxProductVariationEX4200 5 }

jnxChassisEX4200          OBJECT IDENTIFIER ::= { jnxChassis          31 }
  jnxEX4200RE0             OBJECT IDENTIFIER ::= { jnxChassisEX4200 1 }
  jnxEX4200RE1             OBJECT IDENTIFIER ::= { jnxChassisEX4200 2 }
  jnxSlotEX4200            OBJECT IDENTIFIER ::= { jnxSlot            31 }
    jnxEX4200SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotEX4200 1 }
      jnxEX4200SlotPower   OBJECT IDENTIFIER ::= { jnxEX4200SlotFPC 1 }
      jnxEX4200SlotFan     OBJECT IDENTIFIER ::= { jnxEX4200SlotFPC 2 }

jnxMediaCardSpaceEX4200   OBJECT IDENTIFIER ::= { jnxMediaCardSpace 31 }
  jnxEX4200MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceEX4200 1 }
}

jnxModuleEX4200           OBJECT IDENTIFIER ::= { jnxModule          31 }
  jnxEX4200FPC             OBJECT IDENTIFIER ::= { jnxModuleEX4200 1 }
    jnxEX4200Power         OBJECT IDENTIFIER ::= { jnxEX4200FPC 1 }
    jnxEX4200Fan           OBJECT IDENTIFIER ::= { jnxEX4200FPC 2 }

```

## MIB Objects for the SRX 5600 Services Gateway

---

The Chassis MIB objects for the SRX 5600 Services Gateway include:

```
jnxProductLineSRX5600      OBJECT IDENTIFIER ::= { jnxProductLine      28 }
jnxProductNameSRX5600      OBJECT IDENTIFIER ::= { jnxProductName      28 }
jnxProductModelSRX5600     OBJECT IDENTIFIER ::= { jnxProductModel     28 }
jnxProductVariationSRX5600 OBJECT IDENTIFIER ::= { jnxProductVariation 28 }
jnxChassisSRX5600          OBJECT IDENTIFIER ::= { jnxChassis          28 }

jnxSlotSRX5600             OBJECT IDENTIFIER ::= { jnxSlot             28 }
jnxSRX5600SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotSRX5600 1 }
jnxSRX5600SlotHM          OBJECT IDENTIFIER ::= { jnxSlotSRX5600 2 }
jnxSRX5600SlotPower        OBJECT IDENTIFIER ::= { jnxSlotSRX5600 3 }
jnxSRX5600SlotFan          OBJECT IDENTIFIER ::= { jnxSlotSRX5600 4 }
jnxSRX5600SlotCB           OBJECT IDENTIFIER ::= { jnxSlotSRX5600 5 }
jnxSRX5600SlotFPB          OBJECT IDENTIFIER ::= { jnxSlotSRX5600 6 }

jnxMediaCardSpaceSRX5600   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   28
}
jnxSRX5600MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX5600 1
}

jnxMidplaneSRX5600         OBJECT IDENTIFIER ::= { jnxBackplane         28 }
```

## MIB Objects for the SRX 5800 Services Gateway

---

The Chassis MIB objects for the SRX 5800 Services Gateway include:

```
jnxProductLineSRX5800      OBJECT IDENTIFIER ::= { jnxProductLine      26 }
jnxProductNameSRX5800      OBJECT IDENTIFIER ::= { jnxProductName      26 }
jnxProductModelSRX5800     OBJECT IDENTIFIER ::= { jnxProductModel     26 }
jnxProductVariationSRX5800 OBJECT IDENTIFIER ::= { jnxProductVariation 26 }
jnxChassisSRX5800          OBJECT IDENTIFIER ::= { jnxChassis          26 }

jnxSlotSRX5800             OBJECT IDENTIFIER ::= { jnxSlot             26 }
jnxSRX5800SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotSRX5800 1 }
jnxSRX5800SlotHM          OBJECT IDENTIFIER ::= { jnxSlotSRX5800 2 }
jnxSRX5800SlotPower        OBJECT IDENTIFIER ::= { jnxSlotSRX5800 3 }
jnxSRX5800SlotFan          OBJECT IDENTIFIER ::= { jnxSlotSRX5800 4 }
jnxSRX5800SlotCB           OBJECT IDENTIFIER ::= { jnxSlotSRX5800 5 }
jnxSRX5800SlotFPB          OBJECT IDENTIFIER ::= { jnxSlotSRX5800 6 }

jnxMediaCardSpaceSRX5800   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   26
}
jnxSRX5800MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX5800
1 }

jnxMidplaneSRX5800         OBJECT IDENTIFIER ::= { jnxBackplane         26 }
```





## Chapter 23

# 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/junos92/swconfig-net-mgmt/mib-jnx-dcu.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dcu.txt).



**NOTE:** Class-based filter match conditions are not supported on J-series Services Routers.

This chapter contains the following topics:

- `jnxDCUsTable` on page 387
- `jnxDcuStatsTable` on page 388

## `jnxDCUsTable`

The entries in the `jnxDCUsTable`, whose object identifier is `{jnxDCUTable 1}`, are represented by `jnxDCUsEntry` and are listed in Table 70 on page 387.

**Table 70: `jnxDCUsEntry`**

Object	Object Identifier	Description
<code>jnxDCUSrcIfIndex</code>	<code>jnxDCUsEntry 1</code>	The interface index of the ingress interface
<code>jnxDCUDstClassName</code>	<code>jnxDCUsEntry 2</code>	The destination class name specified in a routing policy and applied to the forwarding table.

**Table 70: jnxDCUsEntry (continued)**

Object	Object Identifier	Description
jnxDCUPackets	jnxDCUsEntry3	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 71 on page 388.

**Table 71: jnxDCUsStatusEntry**

Object	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.
jnxDcuStatsClassName	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.
jnxDcuStatsCIName	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 24

# 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 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 a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bgpmib2.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 on page 389

## jnxBgpM2PrefixCountersTable

jnxBgpM2PrefixCountersTable contains counters associated with a BGP peer.

- JnxBgpM2PrefixCountersEntry on page 389

## JnxBgpM2PrefixCountersEntry

jnxBgpM2PrefixCountersEntry contains information about the prefix counters of a BGP peer, and the objects listed in Table 72 on page 389.

**Table 72: jnxBgpM2PrefixCountersEntry**

Object	Object Identifier	Description
jnxBgpM2PrefixInPrefixes	jnxBgpM2PrefixCountersEntry7	The total number of prefixes received from a peer.

**Table 72: jnxBgpM2PrefixCountersEntry** (continued)

Object	Object Identifier	Description
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 25

# Interpreting the Enterprise-Specific Ping MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-ping.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ping.txt).

For more information on using the Ping MIB and enterprise-specific Ping MIB, see “SNMP Remote Operations” on page 87. 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` on page 391
- `jnxPingResultsTable` on page 395
- `jnxPingProbeHistoryTable` on page 398
- `jnxPingLastTestResultTable` on page 400

## **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`. `jnxpingCtlTable` monitors thresholds; for example, the maximum allowed jitter in the trip time during a test.

- `jnxPingCtlEntry` on page 392

## 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 73 on page 392.

**Table 73: 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	<p>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:</p> <ul style="list-style-type: none"> <li>■ pingCtlIfIndex (see pingCtlTable in the Ping MIB)</li> <li>■ jnxPingCtlIfName</li> <li>■ jnxPingCtlRoutingInstanceName</li> </ul>
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.

**Table 73: jnxPingCtlEntry** (continued)

Object	Object Identifier	Description
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 ( <code>jnxPingResultsMaxRttUs</code> minus <code>jnxPingResultsMinRttUs</code> ). If the measured jitter exceeds this threshold, a <code>jnxPingRttJitterThresholdExceeded</code> trap is sent.
jnxPingCtlEgressTimeThreshold	jnxPingCtlEntry 10	Maximum egress trip time allowed. If this threshold is crossed by any probe, a <code>jnxPingEgressThresholdExceeded</code> trap will be sent. This applies only if the probe type ( <code>pingCtlType</code> ) provides one-way delay measurements. Currently <code>jnxPingIcmpTimeStamp</code> is the only supported probe type with this property.
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 <code>jnxPingEgressStdDevThresholdExceeded</code> trap will be sent. This applies only if the probe type ( <code>pingCtlType</code> ) provides one-way delay measurements. The <code>jnxPingIcmpTimeStamp</code> 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 ( <code>jnxPingResultsMaxSrcDstt</code> minus <code>jnxPingResultsMinSrcDstt</code> ). If the measured jitter exceeds this threshold, a <code>jnxPingEgressJitterThresholdExceeded</code> trap will be sent. This applies only if the probe type ( <code>pingCtlType</code> ) provides one-way delay measurements. The <code>jnxPingIcmpTimeStamp</code> 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 <code>jnxPingIngressThresholdExceeded</code> trap will be sent. This applies only if the probe type ( <code>pingCtlType</code> ) provides one-way delay measurements. The <code>jnxPingIcmpTimeStamp</code> is the only supported probe type with this property.

**Table 73: jnxPingCtlEntry** (continued)

Object	Object Identifier	Description
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 <b>jnxPingIngressStddevThresholdExceeded</b> trap will be sent. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. Currently <b>jnxPingIcmpTimeStamp</b> 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 ( <b>jnxPingResultsMaxDstSrcT</b> minus <b>jnxPingResultsMinDstSrcT</b> ). If the measured jitter exceeds this threshold, a <b>jnxPingIngressJitterThresholdExceeded</b> trap will be sent. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. The <b>jnxPingIcmpTimeStamp</b> is the only supported probe type with this property.
jnxPingCtlTrapGeneration	jnxPingCtlEntry 16	<p>The value of this object determines when and if to generate a notification for this entry.</p> <p><b>rttThreshold(0)</b>—Generate a <b>jnxPingRttThresholdExceeded</b> notification when the configured RTT threshold is exceeded.</p> <p><b>rttStdDevThreshold(1)</b>—Generate a <b>jnxPingRttStdDevThresholdExceeded</b> notification when the configured RTT standard deviation threshold is exceeded.</p> <p><b>rttJitterThreshold(2)</b>—Generate a <b>jnxPingRttJitterThresholdExceeded</b> notification when the configured RTT jitter threshold is exceeded.</p> <p><b>egressThreshold(3)</b>—Generate a <b>jnxPingEgressThresholdExceeded</b> notification when the configured egress threshold is exceeded. This applies only if the probe type supports one-way measurements.</p>



**Table 73: jnxPingCtlEntry** (continued)

Object	Object Identifier	Description
		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 on page 395

## jnxpingResultsEntry

The jnxPingResultsEntry objects are listed in Table 74 on page 396.

**Table 74: 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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.

**Table 74: jnxPingsResultsEntry (continued)**

Object	Object Identifier	Description
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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) 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.

**Table 74: jnxPingsResultsEntry (continued)**

Object	Object Identifier	Description
jnxPingResultsJitterEgressUs	jnxPingResultsEntry 18	The jitter of the egress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) 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.
jnxPingResultsOwnerIndex	jnxPingResultsEntry 22	The first index. It has the same value as <b>pingCtlOwnerIndex</b> and is provided for applications that are unable to obtain the value of <b>pingCtlOwnerIndex</b> from the instance portion of the object identifiers belonging to this table.
jnxPingResultsTestName	jnxPingResultsEntry 23	The other index. It has the same value as <b>pingCtlTestName</b> and is provided for applications that are unable to obtain the value of <b>pingCtlTestName</b> from the instance portion of the object identifiers belonging to this table.

## jnxPingProbeHistoryTable

jnxpingProbeHistoryTable, whose object identifier is **jnxPingObjects 4**, contains the history of all ping tests.

- jnxPingProbeHistoryEntry on page 398

## jnxPingProbeHistoryEntry

The jnxPingProbeHistoryEntry objects are listed in Table 75 on page 399.

**Table 75: 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.
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 ( <code>pingCtlType</code> ) 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 ( <code>pingCtlType</code> ) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.

**Table 75: jnxPingProbeHistoryEntry (continued)**

Object	Object Identifier	Description
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 ( <b>pingCtlType</b> ) 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 ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, the value is irrelevant and will return 0.

## **jnxPingLastTestResultTable**

jnxPingLastTestResultTable, whose object identifier is **jnxPingObjects 5**, contains the results of the last completed ping tests. Entries corresponding to a test are created only after completion of the first test. This is useful when you want to ensure that a test has been completed before collecting test results.

- jnxPingLastTestResultEntry on page 400

## **jnxPingLastTestResultEntry**

The jnxPingLastTestResultEntry objects are listed in Table 76 on page 401.

**Table 76: jnxPingLastTestResultEntry**

Object	Object Identifier	Description
jnxPingLastTestResultProbeResponses	jnxPingLastTestResultEntry 1	The number of responses received in the most recently completed test.
jnxPingLastTestResultSentProbes	jnxPingLastTestResultEntry 2	The number of probes sent in the most recently completed test.
jnxPingLastTestResultSumRttUs	jnxPingLastTestResultEntry 3	The sum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinRttUs	jnxPingLastTestResultEntry 4	The minimum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMaxRttUs	jnxPingLastTestResultEntry 5	The maximum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultAvgRttUs	jnxPingLastTestResultEntry 6	The average of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultStdDevRttUs	jnxPingLastTestResultEntry 7	The standard deviation of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinEgressUs	jnxPingLastTestResultEntry 8	The minimum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <code>pingCtlType</code> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.

**Table 76: jnxPingLastTestResultEntry** (continued)

Object	Object Identifier	Description
jnxPingLastTestResultMaxEgressUs	jnxPingLastTestResultEntry 9	The maximum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultAvgEgressUs	jnxPingLastTestResultEntry 10	The average of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultStddevEgressUs	jnxPingLastTestResultEntry 11	The standard deviation of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultMinIngressUs	jnxPingLastTestResultEntry 12	The minimum of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.



**Table 76: jnxPingLastTestResultEntry** (continued)

Object	Object Identifier	Description
jnxPingLastTestResultMaxIngressUs	jnxPingLastTestResultEntry 13	The maximum of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultAvgIngressUs	jnxPingLastTestResultEntry 14	The average of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultStddevIngressUs	jnxPingLastTestResultEntry 15	The standard deviation of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultPeakToPeakJitterRttUs	jnxPingLastTestResultEntry 16	The difference between the minimum and maximum delays over the course of the last completed test, in microseconds.

**Table 76: jnxPingLastTestResultEntry** (continued)

Object	Object Identifier	Description
jnxPingLastTestResultPeakToPeakJitterEgressUs	jnxPingLastTestResultEntry 17	The difference between the minimum and maximum egress trip delays over the course of the last completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultPeakToPeakJitterIngressUs	jnxPingLastTestResultEntry 18	The difference between the minimum and maximum ingress trip delays over the course of the last completed test, in microseconds. This applies only if the probe type ( <b>pingCtlType</b> ) provides one-way delay measurements. For all other probe types, their values are irrelevant and will return 0.
jnxPingLastTestResultTime	jnxPingLastTestResultEntry 19	The time the last test was completed.

## Chapter 26

# Interpreting the Enterprise-Specific Traceroute MIB

The enterprise-specific 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 enterprise-specific Traceroute MIB. For a downloadable version of the Traceroute MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-traceroute.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-traceroute.txt).

For more information on using the Traceroute MIB and enterprise-specific Traceroute MIB, see “SNMP Remote Operations” on page 87.

This chapter contains the following topic:

- `jnxTraceRouteCtlTable` on page 405

## **`jnxTraceRouteCtlTable`**

The `jnxTraceRouteCtlTable`, whose object identifier is `{jnxTraceRouteObjects 2}`, defines the `jnxTraceRoute` control table for providing enterprise-specific options to the corresponding `traceRouteCtlEntry`.

- `jnxTraceRouteCtlEntry` on page 405

## **`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 77 on page 405.

**Table 77: `jnxTraceRouteCtlTable`**

Object	Object Identifier	Description
<code>jnxTRCtlOwnerIndex</code>	<code>jnxTraceRouteCtlEntry 1</code>	The first index. It is identical to the <code>jnxTraceRouteCtlOwnerIndex</code> of the corresponding <code>jnxTraceRouteCtlEntry</code> in the <code>jnxTraceRouteCtlTable</code> .

**Table 77: jnxTraceRouteCtlTable** (continued)

Object	Object Identifier	Description
jnxTRCtlTestName	jnxTraceRouteCtlEntry 2	The other index. It is identical to the <code>jnxTraceRouteCtlTestName</code> of the corresponding <code>jnxTraceRouteCtlEntry</code> in the <code>jnxTraceRouteCtlTable</code> .
jnxTRCtlIfName	jnxTraceRouteCtlEntry 3	<p>Specifies the name of the outgoing interface for traceroute probes. This is the name-based complement to <code>traceRouteCtlIfIndex</code>. 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.</p> <p>The precedence order is as follows:</p> <ul style="list-style-type: none"> <li>■ <code>traceRouteCtlIfIndex</code> (see <code>traceRouteCtlTable</code> in the Traceroute MIB)</li> <li>■ <code>jnxTRCtlIfName</code></li> <li>■ <code>jnxTRCRoutingInstanceName</code></li> </ul>
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 <code>[edit routing-instances]</code> hierarchy level of the JUNOS configuration.

## Chapter 27

# Interpreting the Enterprise-Specific RMON Events and Alarms MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-rmon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rmon.txt).

For more information on RMON alarms and events, see “Configuring RMON Alarms and Events” on page 221.

This chapter contains the following topics:

- `jnxRmonAlarmTable` on page 407
- RMON Event and Alarm Traps on page 409

## **`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 78 on page 407.

**Table 78: `jnxRmonAlarmEntry`**

Object	Object Identifier	Description
<code>jnxRmonAlarmGetFailCnt</code>	<code>jnxRmonAlarmEntry 1</code>	Represents the number of times the internal <code>Get</code> request for the variable monitored by this entry has failed.
<code>jnxRmonAlarmGetFailTime</code>	<code>jnxRmonAlarmEntry 2</code>	Represents the value of <code>sysUpTime</code> when an internal <code>Get</code> request for the variable monitored by this entry last failed.

**Table 78: jnxRmonAlarmEntry (continued)**

Object	Object Identifier	Description
jnxRmonAlarmGetFailReason	jnxRmonAlarmEntry 3	<ul style="list-style-type: none"> <li>Represents the reason an internal <b>Get</b> request for the variable monitored by this entry last failed. This object contains the following values:</li> <li><b>other (1)</b>—An error was encountered that does not fit into one of the currently defined categories.</li> <li><b>noError (2)</b>—<b>Get</b> request processed successfully.</li> <li><b>noSuchObject (3)</b>—Requested object not available.</li> <li><b>outOfView (4)</b>—Requested object instance out of MIB view.</li> <li><b>noSuchInstance (5)</b>—Requested object instance not available.</li> <li><b>badReqId (6)</b>—Unexpected request ID encountered while processing <b>Get</b> request.</li> <li><b>oidMatchErr (7)</b>—Unexpected object ID encountered while processing <b>Get</b> request.</li> <li><b>oidBindErr (8)</b>—Unable to bind object ID to <b>Get</b> request PDU.</li> <li><b>createPktErr (9)</b>—Unable to create <b>Get</b> request PDU.</li> <li><b>badObjType (10)</b>—Unexpected object type encountered while processing <b>Get</b> request.</li> </ul>
jnxRmonAlarmGetOkTime	jnxRmonAlarmEntry 4	Represents the value of <b>sysUpTime</b> when an internal <b>Get</b> request for the variable monitored by this entry succeeded and the entry left the <b>getFailure</b> state.
jnxRmonAlarmState	jnxRmonAlarmEntry 5	<p>Represents the current state of this RMON alarm entry. This object contains the following values:</p> <ul style="list-style-type: none"> <li><b>unknown (1)</b>—Alarm entry unknown</li> <li><b>underCreation (2)</b>—Alarm entry not activated</li> <li><b>active (3)</b>—Alarm entry active and within thresholds</li> <li><b>startup (4)</b>—Alarm entry still waiting for first value</li> <li><b>risingThreshold (5)</b>—Alarm entry has crossed the rising threshold.</li> <li><b>fallingThreshold (6)</b>—Alarm entry has crossed the falling threshold</li> <li><b>getFailure (7)</b>—Alarm entry internal <b>Get</b> request failed.</li> </ul>

## RMON Event and Alarm Traps

---

The following traps send notifications when there is a problem with RMON alarm processing and are listed in Table 79 on page 409.

**Table 79: RMON Event and Alarm Traps**

Trap	Object Identifier	Description
jnxRmonAlarmGetFailure	jnxRmonTrapPrefix 1	Generated when the <b>Get</b> request for an alarm variable returns an error. The specific error is identified by <b>jnxRmonAlarmGetFailReason</b> .
jnxRmonGetOk	jnxRmonTrapPrefix 2	Generated when the <b>Get</b> request for an alarm variable is successful. This trap is only sent after previous attempts are unsuccessful.





## Chapter 28

# Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-rpf.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpf.txt).

This chapter discusses the following topic:

- jnxRpfStatsTable on page 411

## jnxRpfStatsTable

The **jnxRpfStatsTable**, whose object identifier is {jnxRpfStats 1}, provides a list of RPF entries in table format.

- jnxRpfStatsEntry on page 411

## jnxRpfStatsEntry

The **jnxRpfStatsEntry**, whose object identifier is {jnxRpfStatsTable 1}, has four objects, which are listed in Table 80 on page 411.

**Table 80: 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 29

# Interpreting the Enterprise-Specific Source Class Usage MIB

The enterprise-specific 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 enterprise-specific 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 enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-scu.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-scu.txt).



**NOTE:** Class-based filter match conditions are not supported on J-series Services Routers.

---

This chapter discusses the following topic:

- `jnxScuStatsTable` on page 413

## **`jnxScuStatsTable`**

---

The `jnxRpfStatsTable`, whose object identifier is `{jnxRpfStats 1}`, provides a list of RPF entries in table format.

- `jnxRpfStatsEntry` on page 413

## **`jnxRpfStatsEntry`**

The `jnxRpfStatsEntry`, whose object identifier is `{jnxRpfStatsTable 1}`, has four objects, which are listed in Table 81 on page 414.

**Table 81: 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 30

# Interpreting the Enterprise-Specific Passive Monitoring MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-pmon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pmon.txt).

This chapter documents only *jnxPMonFlowTable*.

This chapter contains the following topic:

- *jnxPMonFlowTable* on page 415

## ***jnxPMonFlowTable***

---

*jnxPMonFlowTable* has an object identifier of {*jnxPMon 1*}. Its entries are represented by *JnxPMonFlowEntry*, which contains the objects listed in Table 82 on page 416.

**Table 82: 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.
jnxPMonTenSecondAverageFlowsPackets	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 31

# Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-sonet.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sonet.txt).

This chapter discusses the following topic:

- `jnxSonetAlarmsTable` on page 417

## ***jnxSonetAlarmsTable***

The `jnxSonetAlarmsTable`, whose object identifier is `{jnxSonetAlarm 1}`, provides information about alarm status on SONET/SDH physical interfaces.

- `jnxSonetAlarmEntry` on page 417

## ***jnxSonetAlarmEntry***

The `jnxSonetAlarmEntry`, whose object identifier is `{jnxSonetAlarmTable 1}`, has five objects, which are listed in Table 83 on page 417.

**Table 83: `jnxSonetAlarmTable`**

Object	Object Identifier	Description
<code>jnxSonetCurrentAlarms</code>	<code>jnxSonetAlarmEntry 1</code>	Identifies all the active SONET/SDH alarms on this interface.
<code>jnxSonetLastAlarmId</code>	<code>jnxSonetAlarmEntry 2</code>	Identifies the SONET/SDH alarm that most recently was set or cleared.
<code>jnxSonetLastAlarmTime</code>	<code>jnxSonetAlarmEntry 3</code>	The value of <code>sysUpTime</code> when the management subsystem learned of the last alarm event.
<code>jnxSonetLastAlarmDate</code>	<code>jnxSonetAlarmEntry 4</code>	The system date and time when the management subsystem learned of the last alarm event.
<code>jnxSonetLastAlarmEvent</code>	<code>jnxSonetAlarmEntry 5</code>	Indicates whether the last alarm event set a new alarm or cleared an existing alarm.

Table 84 on page 418 provides an example of `jnxSonetAlarmInterface` objects on an M20 router.

**Table 84: `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	sonetLoIAlarm(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	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetBerrSdAlarm(8)	8 days, 4:09:46.21	2002-10-23,14:29:23.0,-7:0	cleared(3)
17	sonetLoIAlarm(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	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.16	2002-10-15,10:21:14.0,-7:0	set(2)
20	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.17	2002-10-15,10:21:14.0,-7:0	set(2)
21	–	sonetLoIAlarm(2)	7 days, 11:15:00.15	2002-10-22,21:34:37.0,-7:0	cleared(3)
22	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLoIAlarm(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	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.07	2002-10-15,10:21:14.0,-7:0	set(2)
25	sonetLoIAlarm(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	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
28	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
29	sonetLoIAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)



## Chapter 32

# Interpreting the Enterprise-Specific SONET APS MIB

The enterprise-specific SONET Automatic Protection Switching (APS) 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/junos92/swconfig-net-mgmt/mib-jnx-sonetaps.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sonetaps.txt).

This chapter discusses the following topics:

- `apsConfigTable` on page 419
- `apsStatusTable` on page 421
- `apsChanConfigTable` on page 424
- `apsChanStatusTable` on page 425

## **apsConfigTable**

`apsConfigTable` lists the APS groups that are configured on the system.

- `apsConfigEntry` on page 419

## **apsConfigEntry**

`apsConfigEntry` objects have read access only and are listed in Table 85 on page 420.

**Table 85: apsConfigTable**

Object	Object Identifier	Description
apsConfigName	apsConfigEntry 1	<p>A text name for the APS group.</p> <p>An entry cannot exist in the active state unless all objects in the entry have an appropriate value. Also, all associated <b>apsChanConfigEntry</b> rows must represent a set of consecutive channel numbers beginning with 0 or 1, depending on the selected architecture.</p>
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	<p>The revertive mode of the APS group.</p> <ul style="list-style-type: none"> <li>■ 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.</li> <li>■ Nonrevertive mode—Traffic remains on the protection line until another switch request is received.</li> </ul>
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 <sup>-5</sup> .
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 <sup>-5</sup> .
apsConfigWaitToRestore	apsConfigEntry 9	<p>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.</p> <p>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, <b>WaitToRestore</b> value (MI_WTRtime: 0..(5)..12 minutes).</p>

**Table 85: apsConfigTable** (continued)

Object	Object Identifier	Description
apsConfigCreationTime	apsConfigEntry 10	The value of <b>sysUpTime</b> 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, <i>Textual Conventions for SMIv2</i> .

## apsStatusTable

apsStatusTable provides status information about configured APS groups.

- apsStatusEntry on page 421

## apsStatusEntry

apsStatusEntry objects have read access only and are listed in Table 86 on page 421.

**Table 86: 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	<p>The current status of the APS group. This object has the following values:</p> <ul style="list-style-type: none"> <li>■ <b>modeMismatch</b>—Modes other than 1 + 1 unidirectional monitor protection line K2 bit 5, which indicates the architecture, and K2 bits 6 through 8, which indicate whether 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.</li> <li>■ <b>channelMismatch</b>—A mismatch between the transmitted K1 channel and the received K2 channel has been detected.</li> </ul>

**Table 86: apsStatusTable** (continued)

Object	Object Identifier	Description
apsStatusCurrent (cont.)	apsStatusEntry 3	<ul style="list-style-type: none"> <li>■ <b>psbf</b>—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.</li> <li>■ <b>feplf</b>—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.</li> <li>■ <b>extraTraffic</b>—Indicates whether extra traffic is currently being accepted on the protection line.</li> <li>■ <b>extraTraffic</b>—Indicates whether extra traffic is currently being accepted on the protection line.</li> </ul>
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 <b>apsStatusDiscontinuityTime</b> .
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 <b>apsStatusDiscontinuityTime</b> .

**Table 86: apsStatusTable** (continued)

Object	Object Identifier	Description
apsStatusPSBFs	apsStatusEntry 6	<p>Counts protection switch byte failure conditions. This condition occurs when either an inconsistent APS byte or an invalid code is detected.</p> <p>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.</p> <p>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.</p> <p>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 <b>apsStatusDiscontinuityTime</b>.</p>
apsStatusFEPLFs	apsStatusEntry 7	<p>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 <b>apsStatusDiscontinuityTime</b>.</p>
apsStatusSwitchedChannel	apsStatusEntry 8	<p>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.</p>
apsStatusDiscontinuityTime	apsStatusEntry 9	<p>The value of <b>sysUpTime</b> 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 <b>Counter32</b> object contained in <b>apsStatusTable</b>. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value.</p>

## apsChanConfigTable

apsChanConfigTable lists the APS channels that have been configured in APS groups.

- [apsChanConfigEntry](#) on page 424

## apsChanConfigEntry

apsChanConfigEntry objects have read access only and are listed in Table 87 on page 424.

**Table 87: 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	<p>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.</p> <p>The values 1 through 14 define a working channel. When an attempt is made to set the corresponding <b>apsConfigRowStatus</b> field to active, the <b>apsChanConfigNumber</b> values of all entries with equal <b>apsChanConfigGroupName</b> 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 <b>inconsistentValue</b> is returned to the <b>apsConfigRowStatus</b> set attempt.</p>

**Table 87: apsChanConfigTable** (continued)

Object	Object Identifier	Description
apsChanConfigIfIndex	apsChanConfigEntry 4	<p>The interface index assigned to a SONET LTE. This is an interface with ifType <b>sonet(39)</b>. The value of this object must be unique among all instances of <b>apsChanConfigIfIndex</b>. In other words, a particular SONET LTE can only be configured in one APS group.</p> <p>This object cannot be set if the <b>apsChanConfigGroupName</b> instance associated with this row is equal to an instance of <b>apsConfigName</b> and the corresponding <b>apsConfigRowStatus</b> 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 <b>apsConfigRowStatus</b> value is equal to <b>notInService</b>. The JUNOS software supports only read access.</p>
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, <i>Textual Conventions for SMIV2</i> .

## apsChanStatusTable

apsChanStatusTable provides APS channel statistics.

- apsChanStatusEntry on page 425

## apsChanStatusEntry

apsChanStatusEntry objects have read access only and are listed in Table 88 on page 426.

**Table 88: apsChanStatusTable**

Object	Object Identifier	Description
apsChanStatusCurrent	apsChanStatusEntry 1	<p>The current state of the port. This object has the following values:</p> <p><b>lockedOut</b>—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.</p> <p><b>sd</b>—A signal degrade condition is in effect.</p> <p><b>sf</b>—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.</p> <p><b>wtr</b>—A wait-to-restore state is in effect.</p>
apsChanStatusSignalDegrades	apsChanStatusEntry 2	<p>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 <b>apsConfigSdBerThreshold</b>. 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 <b>apsChanStatusDiscontinuityTime</b>.</p>
apsChanStatusSignalFailures	apsChanStatusEntry 3	<p>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 <b>apsConfigSfBerThreshold</b>. 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 <b>apsChanStatusDiscontinuityTime</b>.</p>



**Table 88: apsChanStatusTable** (continued)

Object	Object Identifier	Description
apsChanStatusSwitchovers	apsChanStatusEntry 4	<p>When queried with index value <b>apsChanConfigNumber</b> other than 0, this object returns the number of times this channel has switched to the protection line.</p> <p>When queried with index value <b>s</b> 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 <b>apsChanStatusDiscontinuityTime</b>.</p>
apsChanStatusLastSwitchover	apsChanStatusEntry 5	<p>When queried with index value <b>apsChanConfigNumber</b> other than 0, this object returns the value of <b>sysUpTime</b> 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.</p> <p>When queried with index value <b>apsChanConfigNumber</b> set to 0, which is the protection line, this object will return the value of <b>sysUpTime</b> 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.</p>

**Table 88: apsChanStatusTable** (continued)

Object	Object Identifier	Description
apsChanStatusSwitchoverSeconds	apsChanStatusEntry 6	<p>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.</p> <p>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 <code>apsChanStatusDiscontinuityTime</code>. For example, if the value of an instance of <code>apsChanStatusSwitchoverSeconds</code> changes from a non-zero value to zero due to revertive switching being disabled. It is expected that the corresponding value of <code>apsChanStatusDiscontinuityTime</code> is updated to reflect the time of the configuration change.</p>
apsChanStatusDiscontinuityTime	apsChanStatusEntry 7	<p>The value of <code>sysUpTime</code> 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 <code>Counter32</code> object contained in <code>apsChanStatusTable</code>. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value for <code>apsChanStatusEntry</code>.</p>

## Chapter 33

# Interpreting the Enterprise-Specific IPsec Monitoring MIB

The enterprise-specific IPsec Monitoring MIB, whose object identifier is {jnxMibs 22}, provides operational and statistical information related to the IPsec and IKE tunnels on Juniper Networks routing platforms. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-monitor-asp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-monitor-asp.txt).

This chapter discusses the following topics:

- jnxIkeTunnelTable on page 429
- jnxIPSecTunnelTable on page 432
- jnxIPSecSaTable on page 434

### **jnxIkeTunnelTable**

---

The IKE tunnel table (jnxIkeTunnelTable), whose object identifier is {jnxIPSecPhaseOne 1}, is used to monitor the IKE security associations established with the remote peers. The MIB variables in this table are used to display the IKE SA attributes and the SA statistics. There is one entry for each IKE SA present.

The key for this table is the combination of a service set name, remote gateway address, and the IKE tunnel index. The service set name is used from the jnxSpSvcSetTable which is implemented as part of the Services PIC MIB. The SNMP manager uses the jnxSpSvcSetTable to get the service set name and this information can then be used to query the jnxIkeTunnelTable for the given service set.

To get only IKE tunnels specific to a particular remote gateway in a service set, the SNMP manager can specify the corresponding service set name and the remote gateway address in the query.

- jnxIkeTunnelEntry on page 429

### **jnxIkeTunnelEntry**

The jnxIkeTunnelEntry, whose object identifier is {jnxIkeTunnelTable 1}, has 25 objects, which are listed in Table 89 on page 430. Each entry contains attributes associated with an active IPsec phase 1 IKE tunnel.

**Table 89: jnxIkeTunnelTable**

Object	Object Identifier	Description
jnxIkeTunIndex	jnxIkeTunnelEntry 1	Index for the table. The value of the index is a number that begins at 1 and is incremented with each tunnel that is created. When the index number reaches 2,147,483,647 the value wraps back to 1.
jnxIkeTunLocalRole	jnxIkeTunnelEntry 2	The role of the local peer identity. The role can be <b>initiator</b> or <b>responder</b> .
jnxIkeTunNegState	jnxIkeTunnelEntry 3	The state of the current negotiation. The state can be <b>matured</b> or <b>non matured</b> .
jnxIkeTunInitiatorCookie	jnxIkeTunnelEntry 4	Cookie generated by the peer that initiated the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunResponderCookie	jnxIkeTunnelEntry 5	Cookie generated by the peer responding to the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunLocalIdType	jnxIkeTunnelEntry 6	The type of local peer identity. A local peer can be identified by an IP address, a fully-qualified domain name, or a distinguished name.
jnxIkeTunLocalIdValue	jnxIkeTunnelEntry 7	<p>The value of the local peer identity.</p> <ul style="list-style-type: none"> <li>■ If the local peer type is an IP address, then this is the IP address used to identify the local peer.</li> <li>■ If the local peer type is a fully-qualified domain name (if_fqdn), then this is the fully-qualified domain name (FQDN) of the remote peer.</li> <li>■ If the local peer type is a distinguished name (id_dn), then this is the distinguished name of the local peer.</li> </ul>
jnxIkeTunLocalGwAddrType	jnxIkeTunnelEntry 8	The IP address type of the local endpoint (gateway) for the IPSec phase 1 IKE tunnel.
jnxIkeTunLocalGwAddr	jnxIkeTunnelEntry 9	The IP address of the local endpoint (gateway) for the IPSec phase 1 IKE tunnel.
jnxIkeTunLocalCertName	jnxIkeTunnelEntry 10	The name of the certificate used for authentication of the local tunnel endpoint. This object has a valid value only if the negotiated IKE authentication method is something other than a pre-shared key. If the IKE negotiation does not use certificates for authentication, the value is <b>NULL</b> .

**Table 89: jnxIkeTunnelTable** (continued)

Object	Object Identifier	Description
jnxIkeTunRemoteldType	jnxIkeTunnelEntry 11	The type of remote peer identity. A remote peer can be identified by an IP address, a fully-qualified domain name, or a distinguished name.
jnxIkeTunRemoteldValue	jnxIkeTunnelEntry 12	The value of the remote peer identity. <ul style="list-style-type: none"> <li>■ If the remote peer type is an IP address, then this is the IP address used to identify the remote peer.</li> <li>■ If the remote peer type is a fully-qualified domain name (if_fqdn), then this is the fully-qualified domain name (FQDN) of the remote peer.</li> <li>■ If the remote peer type is a distinguished name (id_dn), then this is the distinguished name of the remote peer.</li> </ul>
jnxIkeTunRemoteGwAddrType	jnxIkeTunnelEntry 13	The IP address type of the remote gateway (endpoint) for the IPSec phase 1 IKE tunnel.
jnxIkeTunRemoteGwAddr	jnxIkeTunnelEntry 14	The IP address of the remote gateway (endpoint) for the IPSec phase 1 IKE tunnel.
jnxIkeTunNegoMode	jnxIkeTunnelEntry 15	The negotiation mode of the IPSec phase 1 IKE tunnel.
jnxIkeTunDiffHellmanGrp	jnxIkeTunnelEntry 16	The Diffie Hellman Group used in IPSec phase 1 IKE negotiations.
jnxIkeTunEncryptAlgo	jnxIkeTunnelEntry 17	The encryption algorithm used in IPSec phase 1 IKE negotiations.
jnxIkeTunHashAlgo	jnxIkeTunnelEntry 18	The hash algorithm used in IPSec phase 1 IKE negotiations.
jnxIkeTunAuthMethod	jnxIkeTunnelEntry 19	The authentication method used in IPSec phase 1 IKE negotiations.
jnxIkeTunLifeTime	jnxIkeTunnelEntry 20	The negotiated lifetime (in seconds) of the IPSec phase 1 IKE tunnel.
jnxIkeTunActiveTime	jnxIkeTunnelEntry 21	The length of time (in hundredths of seconds) that the IPSec phase 1 IKE tunnel has been active.
jnxIkeTunInOctets	jnxIkeTunnelEntry 22	The total number of octets received by this IPSec phase 1 IKE security association.
jnxIkeTunInPkts	jnxIkeTunnelEntry 23	The total number of packets received by this IPSec phase 1 IKE security association.
jnxIkeTunOutOctets	jnxIkeTunnelEntry 24	The total number of octets sent by this IPSec phase 1 IKE security association.

**Table 89: jnxIkeTunnelTable (continued)**

Object	Object Identifier	Description
jnxIkeTunOutPkts	jnxIkeTunnelEntry 25	The total number of octets sent by this IPSec phase 1 IKE security association.

## jnxIPSecTunnelTable

The IPSec phase 2 tunnel table (`jnxIPSecTunnelTable`), whose object identifier is `{jnxIPSecPhaseTwo 1}`, is used to monitor the IPSec phase 2 tunnel attributes along with the statistics from the tunnel. There is one entry for each tunnel to the peer security gateway. This table does not contain information on IPSec security associations (SAs) because multiple SAs can be present for each tunnel.

Similar to the IKE tunnel table (`jnxIkeTunnelTable`), the key of this table is a combination of the service set name, remote gateway address, and the IPSec tunnel index. This table can be queried just like the IKE tunnel table.

To get only IPSec tunnels specific to a particular remote gateway in a service set, the SNMP manager can specify the corresponding service set name and the remote gateway address in the query.

- `jnxIPSecTunnelEntry` on page 432

## jnxIPSecTunnelEntry

The `jnxIPSecTunnelEntry`, whose object identifier is `{jnxIPSecTunnelTable 1}`, has 27 objects, which are listed in Table 90 on page 432. Each entry contains attributes associated with an active IPSec phase 2 tunnel.

**Table 90: jnxIPSecTunnelTable**

Object	Object Identifier	Description
jnxIPSecTunIndex	jnxIPSecTunnelEntry 1	Index for the table. The value of the index is a number that begins at 1 and is incremented with each tunnel that is created. When the index number reaches 2,147,483,647 the value wraps back to 1.
jnxIPSecRuleName	jnxIPSecTunnelEntry 2	The name of the rule defined in the IPSec configuration.
jnxIPSecTermName	jnxIPSecTunnelEntry 3	The name of the term configured under the IPSec rule.
jnxIPSecTunLocalGwAddrType	jnxIPSecTunnelEntry 4	The IP address type of the local gateway (endpoint) for the IPSec phase 2 tunnel.
jnxIPSecTunLocalGwAddr	jnxIPSecTunnelEntry 5	The IP address of the local gateway (endpoint) for the IPSec phase 2 tunnel.

**Table 90: jnxIPSecTunnelTable** (continued)

Object	Object Identifier	Description
jnxIPSecTunRemoteGwAddrType	jnxIPSecTunnelEntry 6	The IP address type of the remote gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunRemoteGwAddr	jnxIPSecTunnelEntry 7	The IP address of the remote gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunLocalProxyId	jnxIPSecTunnelEntry 8	The identifier for the local endpoint.
jnxIPSecTunRemoteProxyId	jnxIPSecTunnelEntry 9	The identifier for the remote endpoint.
jnxIPSecTunKeyType	jnxIPSecTunnelEntry 10	The type of key used by the IPsec phase 2 tunnel. The key type can be <b>IKE negotiated</b> or <b>Manually installed</b> .
jnxIPSecRemotePeerType	jnxIPSecTunnelEntry 11	The type of the remote peer gateway (endpoint). If the remote peer's IP address is known beforehand, the type is <b>static</b> . If the IP address is not known beforehand, the type is <b>dynamic</b> .
jnxIPSecTunMtu	jnxIPSecTunnelEntry 12	The maximum transmission unit (MTU) value of the IPsec phase 2 tunnel.
jnxIPSecTunOutEncryptedBytes	jnxIPSecTunnelEntry 13	The number of bytes encrypted by the IPsec phase 2 tunnel.
jnxIPSecTunOutEncryptedPkts	jnxIPSecTunnelEntry 14	The number of packets encrypted by the IPsec phase 2 tunnel.
jnxIPSecTunInDecryptedBytes	jnxIPSecTunnelEntry 15	The number of bytes decrypted by the IPsec phase 2 tunnel.
jnxIPSecTunInDecryptedPkts	jnxIPSecTunnelEntry 16	The number of packets decrypted by the IPsec phase 2 tunnel.
jnxIPSecTunAHInBytes	jnxIPSecTunnelEntry 17	The number of incoming bytes authenticated using the authentication header (AH) by the IPsec phase 2 tunnel.
jnxIPSecTunAHInPkts	jnxIPSecTunnelEntry 18	The number of incoming packets authenticated using the authentication header (AH) by the IPsec phase 2 tunnel.
jnxIPSecTunAHOutBytes	jnxIPSecTunnelEntry 19	The number of outgoing bytes on the IPsec phase 2 tunnel where the AH is applied.
jnxIPSecTunHAOutPkts	jnxIPSecTunnelEntry 20	The number of outgoing packets on the IPsec phase 2 tunnel where the AH is applied.
jnxIPSecTunReplayDropPkts	jnxIPSecTunnelEntry 21	The number of packets dropped by the IPsec phase 2 tunnel because of an anti-replay check failure.

**Table 90: jnxIPSecTunnelTable** (continued)

Object	Object Identifier	Description
jnxIPSecTunAhAuthFails	jnxIPSecTunnelEntry 22	The number of packets received by the IPsec phase 2 tunnel that failed AH authentication.
jnxIPSecTunEspAuthFails	jnxIPSecTunnelEntry 23	The number of packets received by this IPsec phase 2 tunnel that failed ESP authentication.
jnxIPSecTunDecryptFails	jnxIPSecTunnelEntry 24	The number of packets received by this IPsec phase 2 tunnel that failed decryption.
jnxIPSecTunBadHeaders	jnxIPSecTunnelEntry 25	The number of packets received by this IPsec phase 2 tunnel that failed because of bad headers.
jnxIPSecTunBadTrailers	jnxIPSecTunnelEntry 26	The number of packets received by this IPsec phase 2 tunnel that failed because of bad ESP trailers.
jnxIPSecTunDroppedPkts	jnxIPSecTunnelEntry 27	The total number of packets dropped from this IPsec phase 2 tunnel.

## jnxIPSecSaTable

The IPsec phase 2 security association table (**jnxIPSecSaTable**), whose object identifier is {jnxIPSecPhaseTwo 2}, is used to monitor the IPsec SAs present for each tunnel in the IPsec tunnel table (**jnxIPSecTunnelTable**). More than one pair of SAs can be present for each of the IPsec tunnels.

The key for this table is a combination of a service set name, remote gateway address, IPsec tunnel index, and the SA index. While the IPsec tunnel table is queried using the service set name, the SA table can be queried for the IPsec tunnel using the service set name, remote gateway address, and the IPsec tunnel index.

- jnxIPSecSaEntry on page 434

## jnxIPSecSaEntry

The **jnxIPSecSaEntry**, whose object identifier is {jnxIPSecSaTable 1}, has 16 objects, which are listed in Table 91 on page 434. Each entry contains SA components for an active IPsec phase 2 tunnel.

**Table 91: jnxIPSecSaTable**

Object	Object Identifier	Description
jnxIPSecSaProtocol	jnxIPSecSaEntry 1	The index represents the security protocol (AH, ESP, or IPComp) for which the SA was created.



**Table 91: jnxIPSecSaTable** (continued)

Object	Object Identifier	Description
jnxIpSecSaIndex	jnxIpSecSaEntry 2	The index (in the context of the IPsec tunnel) for the SA. The value of the index is a number that begins at 1 and is incremented with each security parameter index (SPI) associated with an IPsec phase 2 tunnel. When the index number reaches 2,147,483,647 the value wraps back to 1.
jnxIpSecSaInSpi	jnxIpSecSaEntry 3	The value of the incoming SPI.
jnxIpSecSaOutSpi	jnxIpSecSaEntry 4	The value of the outgoing SPI.
jnxIpSecSaInAuxSpi	jnxIpSecSaEntry 5	The value of the incoming auxiliary SPI. This object is valid for AH and ESP bundles.
jnxIpSecSaOutAuxSpi	jnxIpSecSaEntry 6	The value of the outgoing auxiliary SPI. This object is valid for AH and ESP bundles.
jnxIpSecSaType	jnxIpSecSaEntry 7	The type of SA (manual or dynamic).
jnxIpSecSaEncapMode	jnxIpSecSaEntry 8	The encapsulation mode used by the IPsec phase 2 tunnel.
jnxIpSecSaLifeSize	jnxIpSecSaEntry 9	The negotiated size (in kilobytes) of the IPsec phase 2 tunnel.
jnxIpSecSaLifeTime	jnxIpSecSaEntry 10	The negotiated lifetime (in seconds) of the IPsec phase 2 tunnel.
jnxIpSecSaActiveTime	jnxIpSecSaEntry 11	The number of seconds the IPsec phase 2 tunnel has been active.
jnxIpSecSaLifeSizeThreshold	jnxIpSecSaEntry 12	The refresh threshold (in kilobytes) of the SA size.
jnxIpSecSaLifeTimeThreshold	jnxIpSecSaEntry 13	The refresh threshold (in seconds) of the SA lifetime.
jnxIpSecSaEncryptAlgo	jnxIpSecSaEntry 14	The algorithm used to encrypt the packets (es-cbc or 3des-cbc).
jnxIpSecSaAuthAlgo	jnxIpSecSaEntry 15	The algorithm used to authenticate the packets (hmac-md5-96 or hmac-sha1-96).
jnxIpSecSaState	jnxIpSecSaEntry 16	The status of the SA. Status can be <b>active</b> (ready for active use) or <b>expiring</b> (any state an SA goes through before being purged).



## Chapter 34

# Interpreting the Enterprise-Specific Ethernet MAC MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-mac.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mac.txt).

This chapter discusses the following topic:

- jnxMacStatsTable on page 437

## jnxMacStatsTable

The jnxMacStatsTable contains a list of MAC statistics for Gigabit Ethernet interfaces.

- jnxMacStatsEntry on page 437

## jnxMacStatsEntry

jnxMacStatsEntry has six objects, which are listed in Table 92 on page 437.

**Table 92: 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.
jnxMacHCOctets	jnxMacStatsEntry 5	The number of total octets transmitted in this VLAN/MAC address.
jnxMacHCOFrames	jnxMacStatsEntry 6	The number of total frames transmitted in this VLAN/MAC address.



## Chapter 35

# Interpreting the Enterprise-Specific Interface MIB

The enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-if-extensions.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-if-extensions.txt).

This chapter discusses the following topics:

- `jnxIfTable` on page 439
- `ifChassisTable` on page 441

## ***jnxIfTable***

`jnxIfTable` lists traffic statistics, input errors, and output errors for an interface.

- `jnxIfEntry` on page 439

## ***jnxIfEntry***

`jnxIfEntry` objects are listed in Table 93 on page 439.

**Table 93: `jnxIfTable`**

Object	Object Identifier	Description
<code>ifIn1SecRates</code>	<code>ifJnxEntry 1</code>	The number of bits per second delivered by this sublayer to its next higher sublayer.
<code>ifIn1SecOctets</code>	<code>ifJnxEntry 2</code>	The number of octets per second delivered by this sublayer to its next higher sublayer.
<code>ifIn1SecPkts</code>	<code>ifJnxEntry 3</code>	The number of packets per second delivered by this sublayer to its next higher sublayer.
<code>ifOut1SecRate</code>	<code>ifJnxEntry 4</code>	The number of bits per second delivered by this sublayer to its next lower sublayer.
<code>ifOut1SecOctets</code>	<code>ifJnxEntry 5</code>	The number of octets per second delivered by this sublayer to its next lower sublayer.

**Table 93: jnxIfTable** (continued)

Object	Object Identifier	Description
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.
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

**Table 93: jnxIfTable** (continued)

Object	Object Identifier	Description
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 on page 441

## ifChassisEntry

ifChassisEntry objects are listed in Table 94 on page 442.

**Table 94: ifChassisTable**

Object	Object Identifier	Description
ifChassisFpc	ifChassisEntry 1	<p>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.</p> <p>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.</p>
ifChassisPic	ifChassisEntry 2	<p>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.</p> <p>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.</p>
ifChassisPort	ifChassisEntry 3	<p>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.</p> <p>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.</p>
ifChassisChannel	ifChassisEntry 4	<p>The channel identifier for the specified interface if it is part of a channelized interface.</p> <p>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.</p>
ifChassisLogicalUnit	ifChassisEntry 5	<p>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.</p> <p>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.</p>



**Table 94: ifChassisTable** (continued)

Object	Object Identifier	Description
ifChassisPicIndex	ifChassisEntry 6	<p>The indexes for the Chassis MIB tables. This is the instance index that keys into <b>jnxContentsTable</b> in the Chassis MIB.</p> <p>For example, the octet string of 8.1.2.0 means a PIC (“8&amp;” 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). In turn, this PIC index can be plugged in by the NMS directly after any MIB objects in the <b>jnxContentsTable</b> obtain 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.”</p>



## Chapter 36

# 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/junos92/swconfig-net-mgmt/mib-jnx-vpn.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vpn.txt).

This chapter discusses the following topics:

- *jnxVpnInfo* on page 445
- *jnxVpnTable* on page 446
- *jnxVpnIfTable* on page 447
- *jnxVpnPwTable* on page 450
- *jnxVpnRTTable* on page 455
- VPN Traps on page 455

## **jnxVpnInfo**

---

*jnxVpnInfo*, whose object identifier is {*jnxVpnMibObjects 1*}, contains information about the number of configured VPNs and active VPNs.

Table 95 on page 446 lists the supported *jnxVpnInfo* objects, VPNs, and circuit connection services.

**Table 95: 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	–	Number of configured VPNs.
jnxVpnActiveVpns	jnxVpnInfo 2	Yes	Yes	Yes	No	Yes	–	Number of active VPNs.
jnxVpnNextIfIndex	jnxVpnInfo 3	–	–	–	–	–	–	Next free VPN interface index.
jnxVpnNextPwIndex	jnxVpnInfo 4	–	–	–	–	–	–	Next free pseudowire index.
jnxVpnNextRTIndex	jnxVpnInfo 5	–	–	–	–	–	–	Next free route target index.

## jnxVpnTable

jnxVpnTable, whose object identifier is jnxVpnMibobjects 2, lists configured VPNs.

- jnxVpnEntry on page 446

## jnxVpnEntry

JnxVpnEntry contains information about a configured VPN with the objects listed in Table 96 on page 446 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 96: 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	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
JnxVpnStorageType	jnxVpnEntry 4	–	–	–	–	–	–	The storage type.
jnxVpnDescription	jnxVpnEntry 5	Yes	Yes	Yes	Yes	No	–	VPN description.
jnxVpnIdentifierType	jnxVpnEntry 6	Yes	Yes	Yes	Yes	No	–	Type of jnxVpnIdentifier.

**Table 96: Supported jnxVpnEntry Objects, VPNs, and Circuit Connection Services** (continued)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIdentifier	jnxVpnEntry 7	Yes	Yes	Yes	Yes	No	–	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	–	No	No	No	No	–	The number of sites configured in the VPN.
jnxVpnActiveSites	jnxVpnEntry 9	–	No	No	No	No	–	The number of active sites in the VPN.
jnxVpnLocalAddresses	jnxVpnEntry 10	No	No	No	No	No	–	The number of addresses learned from the CE device.
jnxVpnTotalAddresses	jnxVpnEntry 11	No	No	No	No	No	–	The total number of addresses in the VPN routing table.
jnxVpnVpnAge	jnxVpnEntry 12	Yes	Yes	Yes	Yes	No	–	The age of the VPN, in hundredths of a second.

## jnxVpnIfTable

The jnxVpnIfTable, whose object identifier is jnxVpnMibObjects 3, lists VPN interfaces.

- jnxVpnIfEntry on page 447

## jnxVpnIfEntry

jnxVpnIfEntry contains information about VPN interfaces, and has the objects listed in Table 97 on page 448. The first three objects (jnxVpnIfVpnType, jnxVpnIfVpnName, and jnxVpnIfIndex) are indexes and are not included in this table.

**Table 97: 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	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
jnxVpnIfStorageType	jnxVpnIfEntry 5	–	–	–	–	–	–	Identifies the storage type for an object.
jnxVpnIfAssociationPw	jnxVpnIfEntry 6	–	Yes	Yes	Yes	No	–	The index of the associated pseudowire. If no index is associated with a pseudowire, the index is 0. A pseudowire 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.

**Table 97: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services** (continued)

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	–	<p>Indicates the protocol running over a VPN interface.</p> <p>This object contains the following values:</p> <ul style="list-style-type: none"> <li>■ other(0)</li> <li>■ frameRelay(1)</li> <li>■ atmAal5(2)</li> <li>■ atmCell(3)</li> <li>■ ethernetVlan(4)</li> <li>■ ethernet(5)</li> <li>■ ciscoHdlc(6)</li> <li>■ ppp(7)</li> <li>■ cem(8)</li> <li>■ atmVcc(9)</li> <li>■ atmVpc(10)</li> <li>■ vpls(11)</li> <li>■ ipInter-working(12)</li> <li>■ snapInter-working(13)</li> <li>■ static(20)</li> <li>■ rip(21)</li> <li>■ ospf(22)</li> <li>■ bgp(23)</li> <li>■ atmTrunkNNI (129)</li> <li>■ atmTrunkUNI (130)</li> </ul>
jnxVpnIfInBandwidth	jnxVpnIfEntry 8	No	No	No	No	No	–	<p>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.</p>

**Table 97: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services** (continued)

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	–	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	–	<p>Status of a monitored VPN interface.</p> <p>This object contains the following values:</p> <ul style="list-style-type: none"> <li>■ unknown(0)</li> <li>■ noLocal-Interface(1)</li> <li>■ disabled(2)</li> <li>■ encapsulationMismatch(3)</li> <li>■ down(4)</li> <li>■ up(5)</li> </ul>

## jnxVpnPwTable

jnxVpnPwTable, whose object identifier is jnxVpnMibObjects 4, lists pseudowire connections.

- jnxVpnPwEntry on page 450

## jnxVpnPwEntry

jnxVpnPwEntry contains pseudowire information about a VPN that is being monitored, and has the objects listed in Table 98 on page 451. The first three objects (jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex) are indexes and are not listed in this table.



**Table 98: 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	–	–	–	–	–	–	Creates, modifies, and deletes a row in this table.
jnxVpnPwStorageType	jnxVpnPwEntry 5	–	–	–	–	–	–	The storage type.
jnxVpnPwAssociatedInterface	jnxVpnPwEntry 6	–	Yes	Yes	Yes	No	–	The VPN index of the interface associated with a pseudowire. If no interface is associated with a pseudowire, 0 is returned.
jnxVpnPwLocalSiteId	jnxVpnPwEntry 7	–	Yes	Yes	Yes	No	–	The local site identifier for a pseudowire. When there is no local site identifier, 0 is returned.
jnxVpnPwRemoteSiteId	jnxVpnPwEntry 8	–	Yes	Yes	Yes	No	–	The remote site identifier. For example, the site at the end of the pseudowire. When there is no remote site identifier, 0 is returned.
jnxVpnRemotetPeldAddrType	jnxVpnPwEntry 9	–	Yes	Yes	Yes	No	–	The remote PE address. For example, the router at the end of the pseudowire.

**Table 98: Supported jnxVpnEntry Objects, VPNs, and Connection Circuit Services** (continued)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross- Connect	Optical VPN	Description
jnxVpnRemotePeldAddress	jnxVpnPwEntry 10	–	Yes	Yes	Yes	No	–	<p>The type of tunnel over which the pseudowire is carried. If several pseudowires can be carried in one tunnel, each pseudowire is identified by the multiplexer or demultiplexer within a tunnel.</p> <p>This object can contain the following values:</p> <ul style="list-style-type: none"> <li>■ static(1)</li> <li>■ gre(2)</li> <li>■ l2tpv3(3)</li> <li>■ ipSec(4)</li> <li>■ ldp(5)</li> <li>■ rsvpTe(6)</li> <li>■ crLdp(7)</li> </ul>
jnxVpnPwTunnelType	jnxVpnPwEntry 11	–	Yes	Yes	Yes	No	–	The type of tunnel over which the pseudowire is carried.
jnxVpnPwTunnelName	jnxVpnPwEntry 12	–	Yes	Yes	Yes	No	–	The name of the tunnel over which a pseudowire is carried.
jnxVpnPwReceiveDemux	jnxVpnPwEntry 13	–	Yes	Yes	Yes	No	–	The demultiplexer value that identifies received packets associated with this pseudowire.

**Table 98: Supported jnxVpnEntry Objects, VPNs, and Connection Circuit Services** (continued)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwTransmitDemux	jnxVpnPwEntry 14	–	Yes	Yes	Yes	No	–	The demultiplexer value that identifies the transmitted packets associated with this pseudowire.
jnxVpnPwStatus	jnxVpnPwEntry 15	–	Yes	Yes	Yes	No	–	<p>The status of the pseudowire.</p> <p>This object can have the following values:</p> <ul style="list-style-type: none"> <li>■ unknown(0)</li> <li>■ down(1)</li> <li>■ up(2)</li> </ul>
jnxVpnPwTunnelStatus	jnxVpnPwEntry 16	–	No	No	No	No	–	The status of the PE-to-PE tunnel over which the pseudowire is carried.
jnxVpnPwRemoteSiteStatus	jnxVpnPwEntry 17	–	No	No	No	No	–	<p>The interface status at the remote end of the pseudowire.</p> <p>This object can have the following values:</p> <ul style="list-style-type: none"> <li>■ unknown(0)</li> <li>■ outOf - Range(1)</li> <li>■ down(2)</li> <li>■ up(3)</li> </ul>
jnxVpnPwTimeUp	jnxVpnPwEntry 18	–	Yes	Yes	Yes	No	–	The time, in hundredths of a second, that a pseudowire has been operational.

**Table 98: Supported jnxVpnEntry Objects, VPNs, and Connection Circuit Services** (continued)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwTransitions	jnxVpnPwEntry 19	–	Yes	Yes	Yes	No	–	The number of state transitions (up to down and down to up) that a tunnel has undergone.
jnxVpnPwLastTransition	jnxVpnPwEntry 20	–	Yes	Yes	Yes	No	–	The time, in hundredths of a second, since the last transition occurred in a tunnel.
jnxVpnPwPacketsSent	jnxVpnPwEntry 21	–	No	No	No	No	–	The number of packets sent over a pseudowire.
jnxVpnPwOctetsSent	jnxVpnPwEntry 22	–	No	No	No	No	–	The number of octets sent over a pseudowire.
jnxVpnPwPacketsReceived	jnxVpnPwEntry 23	No	No		No	No	–	The number of packets received over a pseudowire.
jnxVpnPwOctetsReceived	jnxVpnPwEntry 24	No	No		No	No	–	The number of octets received over a pseudowire.
jnxVpnPwLRPacketsSent	jnxVpnPwEntry 25	No	No		No	No	–	The number of packets sent over a pseudowire.
jnxVpnPwLROctetsSent	jnxVpnPwEntry 26	No	No		No	No	–	The number of octets sent over a pseudowire.
jnxVpnPwLRPacketsReceived	jnxVpnPwEntry 27	No	No		No	No	–	The number of packets received over a pseudowire.
jnxVpnPwLROctetsReceived	jnxVpnPwEntry 28	No	No		No	No	–	The number of octets received over a pseudowire.

## jnxVpnRTTable

The jnxVpnRTTable, whose object identifier is jnxVpnMibObjects 4, contains route targets for a VPN.

- jnxVpnRTEntry on page 455

## jnxVpnRTEntry

jnxVpnRTEntry lists route targets for a given VPN, and has the objects listed in Table 99 on page 455. The first three objects (jnxVpnRTVpnType, jnxVpnRTVpnName, and jnxVpnRTIndex) are indexes and are not listed in this table.

**Table 99: Supported jnxVpnRTEntry Objects, VPNs, and Circuit Connection Services**

Object	ObjectIdentifier	Layer 3 VPN	Layer 2 VPN	VPLS	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRTRowStatus	jnxVpnRTEntry 4	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
jnxVpnRTStorageType	jnxVpnRTEntry 5	–	–	–	–	–	–	Identifies the storage type for an object.
jnxVpnRTType	jnxVpnRTEntry 6	Yes	Yes	Yes	–	No	–	The type of the following route target. The type can be routeTarget[012] or none.
jnxVpnRT	jnxVpnRTEntry 7	Yes	Yes	Yes	–	No	–	The VPN route target. If jnxVpnRTType is none, the value must be all zeros.
jnxVpnRTFunction	jnxVpnRTEntry 8	Yes	Yes	Yes	–	No	–	The route target export distribution type.

## VPN Traps

The enterprise-specific VPN MIB provides traps for monitoring VPNs. Table 100 on page 456 lists supported VPN traps, VPNs, and circuit connection services.

**Table 100: 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	jnxVpnMIBnotifications 1	Yes	Yes	–	Yes	No	–	Indicates that the interface with the index <code>jnxVpnIfIndex</code> belonging to the <code>jnxVpnIfVpnName</code> of type <code>jnxVpnIfVpnType</code> went up.
jnxVpnIfDown	jnxVpnMIBnotifications 2	Yes	Yes	–	Yes	No	–	Indicates that the interface with index <code>jnxVpnIfIndex</code> belonging to <code>jnxVpnIfVpnName</code> of type <code>jnxVpnIfVpnType</code> went down.
jnxVpnPwUp	jnxVpnMIBnotifications 3	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with the index <code>jnxVpnPwIndex</code> belonging to <code>jnxVpnPwVpnName</code> of type <code>jnxVpnPwVpnType</code> went up.
jnxVpnPwDown	jnxVpnMIBnotifications 4	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with index <code>jnxVpnPwIndex</code> belonging to <code>jnxVpnPwVpnName</code> of type <code>jnxVpnPwVpnType</code> went down.

## Chapter 37

# 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/junos92/swconfig-net-mgmt/mib-jnx-coll.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/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 on page 457
- jnxCollPicIfTable on page 458
- jnxCollFileTable on page 459

## jnxCollGlobalStats

jnxCollGlobalStats provides statistics on all the router's Monitoring Services PICs and has the objects listed in Table 101 on page 457.

**Table 101: 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 on page 458

## jnxCollPicEntry

JnxCollPicEntry contains information about a Monitoring Services PIC. Each Monitoring Services PIC contains one interface and is identified by *lIndex*. It has objects listed in Table 102 on page 458.

**Table 102: jnxCollPicEntry**

Object	Object Identifier	Description
<b>File Statistics</b>		
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 transferred to the FTP server or when there is not enough memory.
<b>Record Statistics</b>		
jnxCollPicIfProcRecords	jnxCollPicIfEntry 10	The number of flow records processed by a Monitoring Services PIC.
jnxCollPicIfProcRecordsRate	jnxCollPicIfEntry 11	The number of flow records processed per second during the current 10-second interval.
jnxCollPicIfPeakProcRecordsRate	jnxCollPicIfEntry 12	The peak number of flow records processed.
<b>Memory Statistics</b>		
jnxCollPicIfMemoryUsed	jnxCollPicIfEntry 13	The amount of memory used, in bytes, by a Monitoring Services PIC.



**Table 102: jnxCollPicEntry** (continued)

Object	Object Identifier	Description
<b>File Statistics</b>		
jnxCollPicIfMemoryFree	jnxCollPicIfEntry 14	The amount of free memory, in bytes, on a Monitoring Services PIC.
<b>FTP Statistics</b>		
jnxCollPicIfFtpBytes	jnxCollPicIfEntry 15	The number of bytes transferred using FTP by a Monitoring Services PIC.
jnxCollPicIfFtpByteRate	jnxCollPicIfEntry 16	The number of bytes per second transferred using FTP, measured during the current 10-second interval.
jnxCollPicIfPeakFtpByteRate	jnxCollPicIfEntry 17	The peak number of bytes per second transferred using FTP.
jnxCollPicIfFtpFiles	jnxCollPicIfEntry 18	The number of files transferred by a Monitoring Services PIC using FTP.
jnxCollPicIfFtpFileRate	jnxCollPicIfEntry 19	The number of files per second transferred using FTP.
jnxCollPicIfPeakFtpFileRate	jnxCollPicIfEntry 20	The peak number of files per second transferred using FTP.
jnxCollPicIfFtpFailures	jnxCollPicIfEntry 21	The number of FTP transfer failures transferred by a Monitoring Services PIC.
<b>Error State Statistics</b>		
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 <b>sysUpTime</b> when the management subsystem last learned of a change to the <b>jnxCollPicIfCurrentState</b> for a Monitoring Services PIC.
jnxCollPicIfStateChangeDate	jnxCollPicIfEntry 25	The system date and time when the management subsystem last learned of a change to the <b>jnxCollPicIfCurrentState</b> 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: <ul style="list-style-type: none"> <li>■ none(1)</li> <li>■ set(2)</li> <li>■ cleared(3)</li> </ul>

## jnxCollFileTable

jnxCollFileTable contains information about each flow collection services file on the router.

- jnxCollFileEntry on page 460

**jnxCollFileEntry**

jnxCollFileEntry contains information about a single file open on a Monitoring Services PIC, and has the objects listed in Table 103 on page 460.

**Table 103: 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	<p>The current state of this file. This object contains the following values:</p> <ul style="list-style-type: none"> <li>■ unknown(1)</li> <li>■ active(2)—The file is actively receiving flow records.</li> <li>■ wait(3)—The file is waiting for export.</li> <li>■ export1(4)—The file is being exported to the primary server.</li> <li>■ export2(5)—The file is being exported to the secondary server.</li> </ul>

## Chapter 38

# 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 enterprise-specific 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/junos92/swconfig-net-mgmt/mib-jnx-sp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-sp.txt).



**NOTE:** The Services PIC MIB is not supported on J-series Services Routers unless the appropriate services license is enabled.

This chapter discusses the following topics:

- *jnxSpSvcSetTable* on page 461
- *jnxSpSvcSetSvcTypeTable* on page 463
- *jnxSpSvcSetIfTable* on page 463
- Service Traps on page 464
- Redundant Interfaces on page 465

## ***jnxSpSvcSetTable***

The *jnxSpSvcSetTable*, whose object identifier is {*jnxSPSvcSet 1*}, provides information about each service set on each Adaptive Services PIC on the router.

- *jnxSpSvcSetEntry* on page 461

## ***jnxSpSvcSetEntry***

The *jnxSpSvcSetEntry*, whose object identifier is {*jnxSpSvcSetTable 1*}, has 11 objects, which are listed in Table 104 on page 462. 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 104: 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: <ul style="list-style-type: none"> <li>■ <b>Unknown</b>—The service type is not known.</li> <li>■ <b>Interface-service</b>—The service is interface based.</li> <li>■ <b>Next-hop-service</b>—The service is next-hop based.</li> </ul>
jnxSpSvcSetMemLimitPktDrops	jnxSpSvcSetEntry 9	Number of packets dropped because the service set exceeded its memory limits (operating in the <b>Red</b> 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` on page 463

## jnxSpSvcSetSvcTypeEntry

The `jnxSpSvcSetSvcTypeEntry`, whose object identifier is `{jnxSpSvcSetSvcTypeTable 1}`, has seven objects, which are listed in Table 105 on page 463. 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 105: jnxSpSvcSetSvcTypeTable**

Object	Object Identifier	Description
<code>jnxSpSvcSetSvcTypeIndex</code>	<code>jnxSpSvcSetSvcTypeEntry 1</code>	An integer used to identify the service type.
<code>jnxSpSvcSetSvcTypeIfName</code>	<code>jnxSpSvcSetSvcTypeEntry 2</code>	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.
<code>jnxSpSvcSetSvcTypeName</code>	<code>jnxSpSvcSetSvcTypeEntry 3</code>	The name of the service type.
<code>jnxSpSvcSetSvcTypeSvcSets</code>	<code>jnxSpSvcSetSvcTypeEntry 4</code>	Number of service sets configured on the Adaptive Services PIC that use this service type.
<code>jnxSpSvcSetSvcTypeMemoryUsage</code>	<code>jnxSpSvcSetSvcTypeEntry 5</code>	Amount of memory used by this service type, expressed in bytes.
<code>jnxSpSvcSetSvcTypePctMemoryUsage</code>	<code>jnxSpSvcSetSvcTypeEntry 6</code>	Amount of memory used by this service type, expressed as a percentage of total memory.
<code>jnxSpSvcSetSvcTypeCpuUtil</code>	<code>jnxSpSvcSetSvcTypeEntry 7</code>	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 Services PIC on the router.

- `jnxSpSvcSetSvcIfEntry` on page 464

## ***jnxSpSvcSetSvcIfEntry***

The `jnxSpSvcSetSvcIfEntry`, whose object identifier is `{jnxSpSvcSetSvcIfTable 1}`, has eight objects, which are listed in Table 106 on page 464. Each entry provides service set information about a single Adaptive Services PIC. Each Adaptive Services PIC is identified by its corresponding index number.

**Table 106: `jnxSpSvcSetSvcIfTable`**

Object	Object Identifier	Description
<code>jnxSpSvcSetSvcIfTableName</code>	<code>jnxSpSvcSetSvcIfEntry 1</code>	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.
<code>jnxSpSvcSetSvcIfSvcSets</code>	<code>jnxSpSvcSetSvcIfEntry 2</code>	The number of service sets configured on the Adaptive Services PIC.
<code>jnxSpSvcSetSvcIfMemoryUsage</code>	<code>jnxSpSvcSetSvcIfEntry 3</code>	Amount of memory used by the Adaptive Services PIC, expressed in bytes.
<code>jnxSpSvcSetSvcIfPctMemoryUsage</code>	<code>jnxSpSvcSetSvcIfEntry 4</code>	Amount of memory used by the Adaptive Services PIC, expressed as a percentage of total memory.
<code>jnxSpSvcSetSvcIfPolMemoryUsage</code>	<code>jnxSpSvcSetSvcIfEntry 5</code>	Amount of policy memory used by the Adaptive Services PIC, expressed in bytes.
<code>jnxSpSvcSetSvcIfPctPolMemoryUsage</code>	<code>jnxSpSvcSetSvcIfEntry 6</code>	Amount of policy memory used by the Adaptive Services PIC, expressed as a percentage of the total.
<code>jnxSpSvcSetSvcIfMemoryZone</code>	<code>jnxSpSvcSetSvcIfEntry 7</code>	<p>The memory usage zone currently occupied by the Adaptive Services PIC. The definitions of each zone are:</p> <ul style="list-style-type: none"> <li>■ Green—All new flows are allowed.</li> <li>■ Yellow—Unused memory is reclaimed. All new flows are allowed.</li> <li>■ Orange—New flows are allowed only for service sets that use less than their equal share of memory.</li> <li>■ Red—No new flows are allowed.</li> </ul>
<code>jnxSpSvcSetSvcIfCpuUtil</code>	<code>jnxSpSvcSetSvcIfEntry 8</code>	<p>Amount of CPU processing used by the Adaptive Services PIC, expressed as a percentage of total CPU usage.</p> <p>J-series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.</p>

## **Service Traps**

The enterprise-specific Services PIC MIB provides traps for monitoring AS PICs. Table 107 on page 465 lists the supported traps.

**Table 107: 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 <code>JnxSpSvcSetIfMemoryZone</code> .
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 <code>JnxSpSvcSetIfMemoryZone</code> .
jnxSpSvcSetCpuExceeded	jnxSPNotificationPrefix 3	Indicates that an Adaptive Services PIC has over 85 % 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 % CPU usage.  This trap is not supported on J-series Services Routers.

## Redundant Interfaces

On M-series routers and T-series routing 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,





## Chapter 39

# Interpreting the Enterprise-Specific Dynamic Flow Capture MIB

The Dynamic Flow Capture (DFC) Physical Interface Card (PIC) forwards passively monitored packets matching a particular filter list to one or more destinations.

The DFC architecture consists of one or more control sources that send requests to a Juniper Networks routing platform to monitor incoming data and then forward any packets that match specific filter criteria to a set of one or more content destinations.

The enterprise-specific DFC MIB, whose object identifier is {jnxMibs 33}, sends the current operational status for each dynamic flow capture PIC. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dfc.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-dfc.txt).



**NOTE:** The dynamic flow capture PIC is supported on M320 and T320 routers and the T640 Internet routing node.

---

This chapter discusses the following topics:

- jnxDfcCSTable on page 467
- jnxDfcCDTable on page 471
- DFC Notification Variables on page 471
- DFC Notification Definitions on page 472

## jnxDfcCSTable

---

The jnxDfcCSTable, whose object identifier is {jnxDfc 1}, provides information about the DFC control source on each dynamic flow capture PIC on the router.

- jnxDfcCSEntry on page 467

## jnxDfcCSEntry

The jnxDfcCSEntry, whose object identifier is {jnxDfcCSTable 1}, has 44 objects, which are listed in Table 108 on page 468.

**Table 108: jnxDfcCSTable**

Object	Object Identifier	Description
jnxDfcCSGrpName	jnxDfcCSEntry 1	The name assigned to a DFC group. A DFC group defines a profile of dynamic flow capture configuration information.
jnxDfcCSId	jnxDfcCSEntry 2	Control source identifier. The control source is a client that monitors electronic data or voice transfer over the network. The control source sends filter requests to the Juniper Networks routing platform using a control protocol. The control source has a unique identifier and an optional list of IP addresses.
jnxDfcCSControlProtocolAddRequests	jnxDfcCSEntry 3	The number of control protocol <b>Add</b> requests received. The <b>Add</b> request specifies new filter criteria to be included in the current filter configuration for a given control source and content destination.
jnxDfcCSCriteriaAdded	jnxDfcCSEntry 4	The number of filter criteria added successfully by the control source.
jnxDfcCSCriteriaAdditionFailed	jnxDfcCSEntry 5	The number of filter criteria <b>Add</b> requests that could not be processed successfully.
jnxDfcCSControlProtocolDeleteRequests	jnxDfcCSEntry 6	The number of control protocol <b>Delete</b> requests received. The <b>Delete</b> request specifies filter criteria to be removed from the current filter configuration for a given control source and content destination.
jnxDfcCSCriteriaDeleted	jnxDfcCSEntry 7	The number of filter criteria deleted successfully as requested by the control source.
jnxDfcCSCriteriaDeletionFailed	jnxDfcCSEntry 8	The number of filter criteria <b>Delete</b> requests that could not be processed successfully.
jnxDfcCSCriteriaDeletedTimeoutIdle	jnxDfcCSEntry 9	The number of criteria deleted by timeout idle.
jnxDfcCSCriteriaDeletedTimeoutTotal	jnxDfcCSEntry 10	The number of criteria deleted by timeout total.
jnxDfcCSCriteriaDeletedPackets	jnxDfcCSEntry 11	The number of criteria deleted by packets.
jnxDfcCSCriteriaDeletedBytes	jnxDfcCSEntry 12	The number of criteria deleted by bytes.
jnxDfcCSControlProtocolRefreshRequests	jnxDfcCSEntry 13	The number of control protocol <b>Refresh</b> requests received. The <b>Refresh</b> request updates the timeout for a particular filter criterion (or a set of filter criteria) for a given control source and content destination.
jnxDfcCSCriteriaRefreshed	jnxDfcCSEntry 14	The number of filter criteria <b>Refresh</b> requests processed successfully as requested by the control source.
jnxDfcCSCriteriaRefreshFailed	jnxDfcCSEntry 15	The number of filter criteria <b>Refresh</b> requests that could not be processed successfully.
jnxDfcCSControlProtocolListRequests	jnxDfcCSEntry 16	The number of control protocol <b>List</b> requests received. The <b>List</b> request returns a list of all criteria that a particular control source has added and are currently active.

**Table 108: jnxDfcCSTable** (continued)

Object	Object Identifier	Description
jnxDfcCSListSuccess	jnxDfcCSEntry 17	The number of <b>List</b> requests processed successfully as requested by the control source.
jnxDfcCSListFailed	jnxDfcCSEntry 18	The number of <b>List</b> requests that could not be processed successfully.
jnxDfcCSControlProtocolNoopRequests	jnxDfcCSEntry 19	The number of control protocol <b>Noop</b> requests received. This request is used to verify the end-to-end connectivity between the control source and the DFC PIC.
jnxDfcCSNoopSuccess	jnxDfcCSEntry 20	The number of <b>Noop</b> requests processed successfully as requested by the control source.
jnxDfcCSNoopFailed	jnxDfcCSEntry 21	The number of <b>Noop</b> requests that could not be processed successfully.
jnxDfcCSDynamicCriteriaActive	jnxDfcCSEntry 22	The number of active dynamic filter criteria.
jnxDfcCSStaticCriteriaActive	jnxDfcCSEntry 23	The number of active static filter criteria.
jnxDfcCSBadRequest	jnxDfcCSEntry 24	The number of <b>Bad</b> requests received.
jnxDfcCSResponseSuccessful	jnxDfcCSEntry 25	The number of successful responses corresponding to the <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , <b>List</b> , and <b>Noop</b> requests sent to the control source.
jnxDfcCSResponseImproperCriteria	jnxDfcCSEntry 26	The number of responses generated because of improper filter criteria included in an <b>Add</b> request.
jnxDfcCSResponseUnknownContentDest	jnxDfcCSEntry 27	The number of responses generated because of an unknown content destination included in an <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , or <b>List</b> request.
jnxDfcCSResponseUnknownControlSrc	jnxDfcCSEntry 28	The number of responses generated because of an unknown control source included in an <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , or <b>List</b> request.
jnxDfcCSResponseUnknownCriteriaId	jnxDfcCSEntry 29	The number of responses generated because of an unknown criteria identifier included in an <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , or <b>List</b> request.
jnxDfcCSResponseImproperTimeout	jnxDfcCSEntry 30	The number of responses generated because of an improper timeout specified in an <b>Add</b> or <b>Refresh</b> request.
jnxDfcCSResponseInvalidAuthentication	jnxDfcCSEntry 31	The number of responses generated because of invalid authentication information included in an <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , <b>List</b> , or <b>Noop</b> request.
jnxDfcCSResponseInvalidSequenceNumber	jnxDfcCSEntry 32	The number of responses generated because of an invalid sequence number included in an <b>Add</b> , <b>Delete</b> , <b>Refresh</b> , <b>List</b> , or <b>Noop</b> request.
jnxDfcCSResponseInternalError	jnxDfcCSEntry 33	The number of responses generated because an internal error occurred on the DFC PIC processing the request.

**Table 108: jnxDfcCSTable** (continued)

Object	Object Identifier	Description
jnxDfcCSNotificationRestart	jnxDfcCSEntry 34	The number of <b>Restart</b> notifications sent to configured notification recipients. A notification is generated when a system failure occurs and all DFC filter criteria are lost.
jnxDfcCSNotificationRollover	jnxDfcCSEntry 35	The number of <b>Rollover</b> notifications sent to configured notification recipients. A notification is generated when a sequence number rollover occurs on the DFC PIC.
jnxDfcCSNotificationNoop	jnxDfcCSEntry 36	The number of <b>Noop</b> notifications sent to configured notification recipients. A notification is generated when the DFC PIC receives a <b>Noop</b> message that includes a <b>SendAsync</b> parameter.
jnxDfcCSNotificationTimeout	jnxDfcCSEntry 37	The number of <b>Timeout</b> notifications sent to configured notification recipients. This notification is generated when a DFC PIC times out a filter criterion (based on any one of its configured timeout parameters) and the criterion contains a <b>SendTimeoutAsync</b> parameter.
jnxDfcCSNotificationCongestion	jnxDfcCSEntry 38	A <b>Congestion</b> notification is generated when the total 10-second average packet forwarding rate (in bits per second) summed over all active filter criteria to a configured content destination exceeds the configured <i>soft</i> limit for the destination. The <b>jnxDfcCSNotificationCongestion</b> object contains the number of <b>Congestion</b> notifications sent to configured notification recipients.
jnxDfcCSNotificationCongestionDelete	jnxDfcCSEntry 39	A <b>Congestion Delete</b> notification is generated when the total 10-second average packet forwarding rate (in bits/second) summed over all active filter criteria to a configured content destination exceeds the configured <i>hard</i> limit for the destination. The <b>jnxDfcCSNotificationCongestionDelete</b> object contains the number of <b>Congestion Delete</b> notifications sent to configured notification recipients.
jnxDfcCSNotificationDuplicatesDropped	jnxDfcCSEntry 40	The number of <b>Duplicated Dropped</b> notifications sent to configured notification recipients. This notification is generated when the configurable <b>Maximum Duplicates</b> parameter has been exceeded and packets matching criteria added by the corresponding control source are dropped.
jnxDfcCSAddRequestRate	jnxDfcCSEntry 41	The request processing rate (in requests processed per second).
jnxDfcCSAddRequestPeakRate	jnxDfcCSEntry 42	The peak request processing rate (in requests processed per second).
jnxDfcCSAggrCriteriaBandwidth	jnxDfcCSEntry 43	Bandwidth (in bits per second).
jnxDfcCSSequenceNumber	jnxDfcCSEntry 44	Protocol sequence number.

## jnxDfcCDTable

The jnxDfcCDTable, whose object identifier is {jnxDfc 2}, provides statistical information for content destinations.

- jnxDfcCDEntry on page 471

## jnxDfcCDEntry

The jnxDfcCDEntry, whose object identifier is {jnxDfcCDTable 1}, has seven objects, which are listed in Table 109 on page 471.

**Table 109: jnxDfcCDTable**

Object	Object Identifier	Description
jnxDfcCDGrpName	jnxDfcCDEntry 1	The name assigned to a DFC group. A DFC group defines a profile of dynamic flow capture configuration information.
jnxDfcCDId	jnxDfcCDEntry 2	Content destination identifier. The DFC router processes the requests from the control sources, creates the filters, monitors incoming data flows, and sends the matched packets to their respective content destinations. Content destinations receive the matched packets from the router.
jnxDfcCDCriteria	jnxDfcCDEntry 3	The number of filter criteria configured for the content destination.
jnxDfcCDByteRate	jnxDfcCDEntry 4	The average data rate (in bytes per second) summed over all active filter criteria configured for a given content destination.
jnxDfcCDMatchedPackets	jnxDfcCDEntry 5	The number of packets that match the filter criteria configured for a content destination.
jnxDfcCDMatchedBytes	jnxDfcCDEntry 6	The number of bytes that match the filter criteria configured for a content destination.
jnxDfcCDCongestionNotification	jnxDfcCDEntry 7	The number of Congestion notifications sent to a configured notification recipient.

## DFC Notification Variables

The enterprise-specific DFC MIB provides notifications for monitoring dynamic flow capture. Table 110 on page 471 lists the supported notification variables.

**Table 110: Supported Notification Variables for the DFC MIB**

Object	Object Identifier	Description
jnxDfcInputPktRate	jnxDfcNotifyVars 1	Data packet rate (in packets per second).
jnxDfcPpsSoftOverloadLowWatermark	jnxDfcNotifyVars 2	Configured lowest value for the data packet rate (in packets per second).

**Table 110: Supported Notification Variables for the DFC MIB (continued)**

Object	Object Identifier	Description
jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotifyVars 3	Configured highest value for the data packet rate (in packets per second).
jnxDfcPpsHardOverloadLowWatermark	jnxDfcNotifyVars 4	Recommended lowest value for the data packet rate (in packets per second).
jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotifyVars 5	Recommended highest value for the data packet rate (in packets per second).
jnxDfcFlowsUsage	jnxDfcNotifyVars 6	Percent (%) usage of the total number of flows.
jnxDfcCriteriaUsage	jnxDfcNotifyVars 7	Percent (%) usage of matching criteria for all filters.
jnxDfcMemSoftOverloadLowWatermark	jnxDfcNotifyVars 8	Configured lowest watermark percent for memory load.
jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotifyVars 9	Configured highest watermark percent for memory load.
jnxDfcFlowLowWatermark	jnxDfcNotifyVars 10	Recommended lowest value for the number of flows allowed.
jnxDfcFlowHighWatermark	jnxDfcNotifyVars 11	Recommended highest value for the number of flows allowed.
jnxDfcCriteriaLowWatermark	jnxDfcNotifyVars 12	Recommended lowest value for the number of criteria allowed.
jnxDfcCriteriaHighWatermark	jnxDfcNotifyVars 13	Recommended highest value for the number of criteria allowed.

## DFC Notification Definitions

Table 111 on page 472 lists the supported notification definitions.

**Table 111: Supported Notification Definitions for the DFC MIB**

Notification Type	Objects	Identifier	Description
jnxDfcSoftPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 1	Notification that occurs when the input packet rate (in packets per second) exceeds the configured limit.
jnxDfcSoftPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 2	Notification that occurs when the input packet rate (in packets per second) returns to below the configured limit.

**Table 111: Supported Notification Definitions for the DFC MIB** *(continued)*

Notification Type	Objects	Identifier	Description
jnxDfcHardPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 3	Notification that occurs when the input packet rate (in packets per second) exceeds the recommended limit.
jnxDfcHardPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 4	Notification that occurs when the input packet rate (in packets per second) returns to below the recommended limit.
jnxDfcSoftMemThresholdExceeded	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 5	Notification that occurs when memory usage exceeds the configured limit.
jnxDfcSoftMemUnderThreshold	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 6	Notification that occurs when memory usage returns to below the configured limit.
jnxDfcHardMemThresholdExceeded	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 7	Notification that occurs when memory usage exceeds the recommended limit.
jnxDfcHardMemUnderThreshold	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 8	Notification that occurs when memory usage returns to below the recommended limit.





## Chapter 40

# Interpreting the Enterprise-Specific Chassis Forwarding MIB

The enterprise-specific Chassis Forwarding MIB, whose object identifier is {jnxMibs 34}, enables J-series Services Routers to fully support the JUNOS health monitor. This MIB extends the scope of health monitoring to include JUNOS forwarding process (fwdd) components on J-series Services Routers. The forwarding process is responsible for most of the packet transmission through a J-series Services Router. The overall performance of the router is largely determined by the effectiveness of the forwarding process.

The JUNOS health monitor uses objects in the Chassis Forwarding MIB to access information about the forwarding process such as microkernel CPU usage and real-time thread CPU usage.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-fwdd.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-fwdd.txt).

This chapter contains the following topic:

- jnxFwddProcess on page 475

## jnxFwddProcess

The object identifier for jnxFwddProcess is {jnxFwdd 1}. This object shows health monitoring statistics for the forwarding process (fwdd) (Table 112 on page 475).

**Table 112: jnxFwddProcess**

Object	Object Identifier	Description
jnxFwddMicroKernelCPUUsage	jnxFwddProcess 1	Percentage of the CPU being used by the forwarding process microkernel. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddRtThreadsCPUUsage	jnxFwddProcess 2	Percentage of the CPU being used by the forwarding process real-time threads. If this information is unavailable or is not applicable, the value is 0 (zero).

**Table 112: jnxFwddProcess** *(continued)*

Object	Object Identifier	Description
jnxFwddHeapUsage	jnxFwddProcess 3	Percentage of heap space being used by the forwarding process. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddDmaMemUsage	jnxFwddProcess 4	Percentage of DMA memory used by the forwarding process. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddUpTime	jnxFwddProcess 5	Forwarding process uptime expressed in terms of system uptime. If this information is unavailable or is not applicable, the value is 0 (zero).

## Chapter 41

# Interpreting the Enterprise-Specific System Log MIB

Event policies can include an action that raises traps for events based on system log messages. This feature enables notification of an SNMP trap-based application when an important system log message occurs. You can convert any system log message (for which there are no corresponding traps) into a trap. This feature is valuable for customers who use network management system traps rather than system log messages to monitor their networks. For more information on converting system log messages into traps, see the *JUNOS Configuration and Diagnostic Automation Guide*.

The enterprise-specific System Log MIB, whose object identifier is {jnxMibs 35}, provides support for this feature.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-syslog.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-syslog.txt).

This chapter contains the following topics:

- jnxSyslogTable on page 477
- jnxSyslogAvTable on page 479

## **jnxSyslogTable**

---

The jnxSyslogTable, whose object identifier is {jnxSyslog 1}, provides information about each system log message generated by the router.

- jnxSyslogEntry on page 477

## **jnxSyslogEntry**

The jnxSyslogEntry, whose object identifier is {jnxSyslogTable 1}, has nine objects, which are listed in Table 113 on page 478. Each entry provides information about a single system log message.

**Table 113: jnxSyslogTable**

Object	Object Identifier	Description
jnxSyslogID	jnxSyslogEntry 1	System log message identifier. This identifier is a numerical value and may not be consecutive. This identifier is also used as the primary index in jnxSyslogAttrValTable.
jnxSyslogEventName	jnxSyslogEntry 2	An octet string that contains the system log event name.
jnxSyslogTimestamp	jnxSyslogEntry 3	Time the message was generated. This value is expressed as <b>DateAndTime</b> .
jnxSyslogSeverity	jnxSyslogEntry 4	<p>Severity of the system log message. The severity value is equal to the value that the system log uses + 1. For example, an emergency message (severity 0 in the system log) has a severity of 1.</p> <p>Severity values include:</p> <ul style="list-style-type: none"> <li>■ emergency (1)</li> <li>■ alert (2)</li> <li>■ critical (3)</li> <li>■ error (4)</li> <li>■ warning (5)</li> <li>■ notice (6)</li> <li>■ info (7)</li> <li>■ debug (8)</li> </ul>
jnxSyslogFacility	jnxSyslogEntry 5	Identifies the facility used to generate the log message. (Table 114 on page 478.)
jnxSyslogProcessID	jnxSyslogEntry 6	Process ID of the process that generated the system log message.
jnxSyslogProcessName	jnxSyslogEntry 7	Process that generated the system log message.
jnxSyslogHostName	jnxSyslogEntry 8	Hostname of the machine that generated the system log.
jnxSyslogMessage	jnxSyslogEntry 9	System log message that was generated.

Table 114 on page 478 lists the facilities that generate system log messages..

**Table 114: Facilities That Generate System Log Messages**

Index	Item	Description
1	kernel	Kernel messages
2	user	User level messages

**Table 114: Facilities That Generate System Log Messages** *(continued)*

Index	Item	Description
3	mail	Mail system
4	daemon	System processes
5	auth	Authorization messages
6	syslog	Messages generated by the system log process (syslogd)
7	lpr	Line printer subsystem
8	news	Network news subsystem
9	uucp	UUCP subsystem
10	cron	Clock process
11	authPriv	Authorization messages
12	ftp	FTP process
13	ntp	NTP subsystem
14	security	Security subsystems (for example, firewall)
15	console	/dev/console output
16	reserved	Reserved for system use
17	local0	–
18	dfc	JUNOS names
19	local2	–
20	firewall	JUNOS names
21	pfe	JUNOS names
22	conflict	JUNOS names
23	change	JUNOS names
24	interact	JUNOS names

## jnxSyslogAvTable

The `jnxSyslogAvTable`, whose object identifier is `{jnxSyslogNotifyVars 2}`, provides information about each system log message generated by the router.

- `jnxSyslogEntry` on page 480

## ***jnxSyslogEntry***

The `jnxSyslogAvEntry`, whose object identifier is `{jnxSyslogAvTable 1}`, has three objects, which are listed in Table 115 on page 480. Each entry provides information about attribute value pairs of system log messages generated by a device.

**Table 115: `jnxSyslogAvTable`**

Object	Object Identifier	Description
<code>jnxSyslogAvIndex</code>	<code>jnxSyslogAvEntry 1</code>	Index for the attribute value pair in the system log message.
<code>jnxSyslogAvAttribute</code>	<code>jnxSyslogAvEntry 2</code>	Attribute of the system log message (identified by <code>jnxSyslogID</code> ).
<code>jnxSyslogAvValue</code>	<code>jnxSyslogAvEntry 3</code>	Value of the attribute (identified by <code>jnxSyslogAvAttribute</code> ).

## Chapter 42

# Interpreting the Enterprise-Specific MPLS LDP MIB

The enterprise-specific MPLS LDF MIB, whose object identifier is {jnxMibs 36}, contains object definitions as described in RFC 3815, *Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)*.

This MIB was supported in earlier releases of JUNOS software as a proprietary LDP MIB (mib-ldpmib.txt). Because the branch used by the proprietary LDP MIB (mib-ldpmib.txt) conflicts with RFC 3812, the proprietary LDP MIB (mib-ldpmib.txt) has been deprecated and replaced by the enterprise-specific MPLS LDP MIB (mib-jnx-mpls-ldp.txt).

For a downloadable version of this MIB, see  
[www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls-ldp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls-ldp.txt).





## Chapter 43

# Interpreting the Enterprise-Specific Packet Forwarding Engine MIB

The enterprise-specific Packet Forwarding Engine (PFE) MIB, whose object identifier is {jnxPfeMibRoot 1}, provides notification statistics for Packet Forwarding Engines. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pfe.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pfe.txt).



**NOTE:** Information provided by this MIB is modeled after information provided by the `show pfe statistics notification` CLI command.

This chapter discusses the following topics:

- jnxPfeNotifyGTable on page 483
- jnxPfeNotifyTypeTable on page 485

## jnxPfeNotifyGTable

jnxPfeNotifyGTable contains global PFE notification statistics.

- jnxPfeNotifyGEntry on page 483

## jnxPfeNotifyGEntry

JnxPfeNotifyGEntry contains notification statistics for each PFE slot. Each entry has objects listed in Table 116 on page 483.

**Table 116: jnxPfeNotifyGEntry**

Object	Object Identifier	Description
jnxPfeNotifyGSlot	jnxPfeNotifyGEntry 1	The slot number for a set of PFE notification statistics.
jnxPfeNotifyGParsed	jnxPfeNotifyGEntry 2	The number of notifications reported by the Packet Forwarding Engine controller, which manages packet forwarding functions.
jnxPfeNotifyGAged	jnxPfeNotifyGEntry 3	The number of notifications dropped because they have expired.

**Table 116: jnxPfeNotifyGIEntry** (continued)

Object	Object Identifier	Description
jnxPfeNotifyGICorrupt	jnxPfeNotifyGIEntry 4	The number of notifications dropped because the notification result format is invalid. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGIIllegal	jnxPfeNotifyGIEntry 5	The number of notifications dropped because the notification has an illegal notification type.
jnxPfeNotifyGISample	jnxPfeNotifyGIEntry 6	The number of sample notifications reported by the Packet Forwarding Engine controller.
jnxPfeNotifyGIGiants	jnxPfeNotifyGIEntry 7	The number of notifications dropped because the notification is larger than the supported direct memory access (DMA) size.
jnxPfeNotifyGITtlExceeded	jnxPfeNotifyGIEntry 8	The number of options/TTL-expired notifications sent to service interfaces as transit packets. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGITtlExcErrors	jnxPfeNotifyGIEntry 9	The number of options/TTL-expired notifications that could not be sent to service interfaces as transit packets because the output interface could not be determined. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGISvcOptAsp	jnxPfeNotifyGIEntry 10	The number of IP options packets sent to a Services PIC.
jnxPfeNotifyGISvcOptRe	jnxPfeNotifyGIEntry 11	The number of IP options packets sent to the Routing Engine.
jnxPfeNotifyGIPostSvcOptOut	jnxPfeNotifyGIEntry 12	The number of notifications re-injected by a Services PIC after processing the associated packets. The PFE will forward these notifications to their actual destination. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGIOptTtlExp	jnxPfeNotifyGIEntry 13	The number of TTL-expired transit packets.
jnxPfeNotifyGIDiscSample	jnxPfeNotifyGIEntry 14	The number of sample notifications dropped because the notifications refer to discarded packets in the PFE.
jnxPfeNotifyGIRateLimited	jnxPfeNotifyGIEntry 15	The number of notification ignored because of PFE software throttling (delaying or refusing requests).
jnxPfeNotifyGIPktGetFails	jnxPfeNotifyGIEntry 16	The number of notifications ignored because DMA memory could not be allocated.
jnxPfeNotifyGIDmaFails	jnxPfeNotifyGIEntry 17	The number of notifications where the DMA of associated packets failed for miscellaneous reasons. This counter is valid for T-series routing platforms only.
jnxPfeNotifyGIDmaTotals	jnxPfeNotifyGIEntry 18	The number of notifications for which the packet DMA completed. This counter is valid for T-series routing platforms only.
jnxPfeNotifyGIUnknowns	jnxPfeNotifyGIEntry 19	The number of notifications that could not be resolved to a known, next hop destination. This counter is valid for T-series routing platforms only.

## jnxPfeNotifyTypeTable

jnxPfeNotifyTypeTable contains information on type-specific PFE notifications for each PFE slot. PFE notification types are listed in Table 117 on page 485.

**Table 117: PFE Notification Types**

Index	Item	Description
1	Illegal	Packets with an invalid notification type
2	Unclassified	Packets that did not have a key lookup performed on them
3	Option	Packets that include L3 options
4	Next Hop	Packets that are destined to the host
5	Discard	Discarded packets sent to the route processor
6	Sample	Unused
7	Redirect	Packets sent back to the interfaces from which they arrived
8	Do Not Fragment	Packets that need to be fragmented, but have a <b>don't fragment (DF)</b> value set.
9	CFDF	Packets that have a DF value set and a maximum transmission unit (MTU) exceeded indicator is triggered.
10	Poison	Packets that have a poisoned next-hop index.

- jnxPfeNotifyTypeEntry on page 485

## jnxPfeNotifyTypeEntry

jnxPfeNotifyTypeEntry contains information about type-specific PFE notifications, and has the objects listed in Table 118 on page 485.

**Table 118: jnxPfeNotifyTypeTable**

Object	Object Identifier	Description
jnxPfeNotifyTypeId	jnxPfeNotifyTypeEntry 1	Identifies the PFE notification type. See Table 117 on page 485 for a list of notification types.
jnxPfeNotifyTypeDescr	jnxPfeNotifyTypeEntry 2	A description of the PFE notification type.
jnxPfeNotifyTypeParsed	jnxPfeNotifyTypeEntry 3	The number of notifications that are parsed successfully.
jnxPfeNotifyTypeInput	jnxPfeNotifyTypeEntry 4	The number of notifications whose associated packets are stored in router processor memory using direct memory access.
jnxPfeNotifyTypeFailed	jnxPfeNotifyTypeEntry 5	The number of notifications that are not parsed successfully.

**Table 118: jnxPfeNotifyTypeTable** (continued)

Object	Object Identifier	Description
jnxPfeNotifyTypeIgnored	jnxPfeNotifyTypeEntry 6	The number of notifications where the notification type in the message does not match any of the valid notification types.

## Chapter 44

# Interpreting the Enterprise-Specific Event MIB

The enterprise-specific Event MIB, whose object identifier is {jnxMibs 37}, defines a generic trap that can be generated using an op script or event policy. This MIB provides the ability to specify a system log string and raise a trap if that system log string is found.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-event.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-event.txt).

This chapter discusses the following topics:

- jnxEventAvTable on page 487
- Notifications for the Event MIB on page 488

## jnxEventAvTable

The jnxEventAvTable, whose object identifier is {jnxEventNotifyVars 2}, provides information about traps generated by op scripts or event policies.

- jnxEventAvEntry on page 487

## jnxEventAvEntry

jnxEventAvEntry, whose object identifier is {jnxEventAvTable 1}, has three objects, which are listed in Table 119 on page 487.

**Table 119: jnxEventAvTable**

Object	Object Identifier	Description
jnxEventAvIndex	jnxEventAvEntry 1	The sequence number of the attribute value pair in the trap generated by a op script or event policy
jnxEventAvAttribute	jnxEventAvEntry 2	The attribute name in the trap generated by an op script or event policy
jnxEventAvValue	jnxEventAvEntry 3	The value of the attribute identified by jnxEventAvAttribute

## Notifications for the Event MIB

---

Table 120 on page 488 lists the supported notifications for the Event MIB.

**Table 120: Supported Notifications for the Event MIB**

Object	Object Identifier	Description
jnxEventTrapDescr	jnxEventNotificationPrefix 1	A notification generated by an op script or event policy. In addition to the jnxEventTrap objects, this notification can include one or more attribute value pairs (identified by jnxEventAvAttribute and jnxEventAvValue).

## Chapter 45

# Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB

The enterprise-specific Bidirectional Forwarding Detection (BFD) MIB, whose object identifier is {jnxBfdMibRoot 1}, sends the current operational status for the transmit interval and detection time of BFD sessions. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bfd.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-bfd.txt).

This chapter discusses the following topics:

- jnxBfdSessTable on page 489
- Notifications for the BFD MIB on page 490

## jnxBfdSessTable

The jnxBfdSessTable, whose object identifier is {jnxBfdObjects 1}, is an extension to the jnxSessTable. It provides information about threshold values for the transmit interval and detection time on BFD sessions.

- jnxBfdSessEntry on page 489

## jnxBfdSessEntry

The jnxBfdSessEntry, whose object identifier is {jnxBfdSessTable 1}, has four objects, which are listed in Table 121 on page 489.

**Table 121: jnxBfdSessTable**

Object	Object Identifier	Description
jnxBfdSessThresTxInterval	jnxBfdSessEntry 1	The threshold value (in microseconds) for the transmit interval. If the current transmit interval value (jnxBfdSessCurrTxInterval) adapts to a value greater than the threshold value, a trap is raised (jnxBfdSessTxIntervalHigh).
jnxBfdSessCurrTxInterval	jnxBfdSessEntry 2	The current transmit interval for the session (in microseconds).

**Table 121: jnxBfdSessTable** (continued)

Object	Object Identifier	Description
jnxBfdSessThreshDectTime	jnxBfdSessEntry 3	The threshold value (in microseconds) for the detection time. If the current detection time value ( <b>jnxBfdSessCurrDectTime</b> ) adapts to a value greater than the threshold value, a trap is raised ( <b>jnxBfdSessDetectionTimeHigh</b> ).
jnxBfdSessCurrDectTime	jnxBfdSessEntry 4	The current detection time for the session (in microseconds).

## Notifications for the BFD MIB

Table 122 on page 490 lists the supported notifications for the BFD MIB.

**Table 122: Supported Notifications for the BFD MIB**

Object	Object Identifier	Description
jnxBfdSessTxIntervalHigh	jnxBfdNotification 1	A notification generated when the threshold value for the transmit interval is configured ( <b>jnxBfdSessThresTxInterval</b> ) and the BFD session transmit interval ( <b>jnxBfdSessCurrTxInterval</b> ) adapts to a value greater than the threshold value. This trap is sent only once, when the threshold is first exceeded. The transmit interval can continue to adapt beyond the threshold value.
jnxBfdSessDetectionTimeHigh	jnxBfdNotification 2	A notification generated when the threshold value for the detection time is configured ( <b>jnxBfdSessThresDectTime</b> ) and the BFD session detection time ( <b>jnxBfdSessCurrDectTime</b> ) adapts to a value greater than the threshold value. This trap is sent only once, when the threshold is first exceeded. The detection time can continue to adapt beyond the threshold value.



## Chapter 46

# Interpreting the Enterprise-Specific Layer 2 Transport Protocol (L2TP) MIB

The enterprise-specific Layer 2 Tunneling Protocol (L2TP) Management Information Base (MIB) enables you to monitor L2TP tunnels and sessions using SNMP. L2TP MIB, whose object identifier is {jnxMibs 49}, provides information related to L2TP tunnels and sessions

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2tp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-l2tp.txt).

This chapter discusses the following topics:

- The L2TP Scalar Status and Statistics Group on page 491
- jnxL2tpTunnelGroupStatsTable on page 492
- jnxL2tpTunnelStatsTable on page 493
- jnxL2tpSessionStatsTable on page 495
- jnxL2tpMlpppBundleStatsTable on page 499

## The L2TP Scalar Status and Statistics Group

Table 123 on page 491 lists the objects in L2TP scalar status and statistics group.

**Table 123: The L2TP Scalar Status and Statistics Group**

Object	Object Identifier	Description
jnxL2tpStatsTotalTunnels	jnxL2tpStats 1	Returns the total number of tunnels that are in established state at the time of querying.
jnxL2tpStatsTotalSessions	jnxL2tpStats 2	Returns the total number of sessions that are in established state at the time of querying.
jnxL2tpStatsControlRxOctets	jnxL2tpStats 3	Returns the number of control channel octets received by the tunnels in established state at the time of querying.
jnxL2tpStatsControlRxPkts	jnxL2tpStats 4	Returns the number of control packets received by the tunnels in established state at the time of querying.
jnxL2tpStatsControlTxOctets	jnxL2tpStats 5	Returns the number of control channel octets that were transmitted to tunnel end points that are in established state at the time of querying.

**Table 123: The L2TP Scalar Status and Statistics Group** (*continued*)

Object	Object Identifier	Description
jnxL2tpStatsControlTxPkts	jnxL2tpStats 6	Returns the number of control packets that were transmitted to the tunnel endpoints that are in established state at the time of querying.
jnxL2tpStatsPayloadRxOctets	jnxL2tpStats 7	Returns the number of payload channel octets that were received on the tunnels that are in established state at the time of querying.
jnxL2tpStatsPayloadRxPkts	jnxL2tpStats 8	Returns the number of payload packets that were received on the tunnels that are in established state at the time of querying.
jnxL2tpStatsPayloadTxOctets	jnxL2tpStats 9	Returns the number of payload channel octets that were transmitted to the tunnel peers that are in established state at the time of querying.
jnxL2tpStatsPayloadTxPkts	jnxL2tpStats 10	Returns the number of payload packets that were transmitted to existing tunnel peers that are in established state at the time of querying.
jnxL2tpStatsErrorTxPkts	jnxL2tpStats 11	Returns the number of packet transmission attempts with errors to the tunnel peers that are in established state at the time of querying.
jnxL2tpStatsErrorRxPkts	jnxL2tpStats 12	Returns the number of packets with errors that were received from the existing tunnel peers that are in established state at the time of querying.

## jnxL2tpTunnelGroupStatsTable

The jnxL2tpTunnelGroupStatsTable, whose object ID is jnxL2tpObjects 2, contains objects that describe the current status and statistics of an L2TP tunnel group.

The jnxL2tpTunnelGroupStatsEntry objects are listed in Table 124 on page 492.

**Table 124: jnxL2tpTunnelGroupStatsTable**

Object	Object Identifier	Description
jnxL2tpTunnelGroupStatsTnlGrpName	jnxL2tpTunnelGroupStatsEntry 1	The name of the particular tunnel group.
jnxL2tpTunnelGroupStatsGatewayAddrType	jnxL2tpTunnelGroupStatsEntry 2	The type of local IP address for L2TP tunnels that are part of the group.
jnxL2tpTunnelGroupStatsGatewayAddr	jnxL2tpTunnelGroupStatsEntry 3	The local IP address for L2TP tunnels that are part of the group.
jnxL2tpTunnelGroupStatsSvcIntfName	jnxL2tpTunnelGroupStatsEntry 4	The name of the service interface that is hosting the tunnel group.
jnxL2tpTunnelGroupStatsTotalTunnels	jnxL2tpTunnelGroupStatsEntry 5	The total number of tunnels that are in the established state at the time of querying.
jnxL2tpTunnelGroupStatsTotalSessions	jnxL2tpTunnelGroupStatsEntry 6	The total number of established sessions in the tunnel group at the time of querying.

## jnxL2tpTunnelStatsTable

The jnxL2tpTunnelStatsTable, whose object ID is jnxL2tpObjects 3, contains objects that describe the current status and statistics of an L2TP tunnel.

A jnxL2tpTunnelStatsEntry represents an L2TP tunnel interface statistics entry and has objects that are listed in Table 125 on page 493.

**Table 125: jnxL2tpTunnelStatsTable**

Object	Object Identifier	Description
jnxL2tpTunnelStatsLocalTID	jnxL2tpTunnelStatsEntry 1	The local tunnel Identifier.
jnxL2tpTunnelStatsServiceInterfac	jnxL2tpTunnelStatsEntry 2	The name of the service interface on which the tunnel is being hosted.
jnxL2tpTunnelStatsTunnelGroup	jnxL2tpTunnelStatsEntry 3	The name of the tunnel group to which the tunnel belongs.
jnxL2tpTunnelStatsRemoteTID	jnxL2tpTunnelStatsEntry 4	The remote tunnel identifier. See RFC 2661, Section 3.1.
jnxL2tpTunnelStatsRemotelpAddrType	jnxL2tpTunnelStatsEntry 5	The type of the remote-end address of the tunnel.
jnxL2tpTunnelStatsRemotelpAddress	jnxL2tpTunnelStatsEntry 6	The remote-end address of the tunnel.
jnxL2tpTunnelStatsRemoteUdpPort	jnxL2tpTunnelStatsEntry 7	The remote-end UDP port of the tunnel.
jnxL2tpTunnelStatsActiveSessions	jnxL2tpTunnelStatsEntry 8	The total number of sessions that are in established state for the tunnel.
jnxL2tpTunnelStatsStat	jnxL2tpTunnelStatsEntry 9	<p>One of the following states for the control tunnel:</p> <ul style="list-style-type: none"> <li>■ <b>cc_responder_accept_new</b>—shows that the tunnel has received and accepted the start control connection request (SCCRQ).</li> <li>■ <b>cc_responder_reject_new</b>—shows that the tunnel has received and rejected the SCCRQ.</li> <li>■ <b>cc_responder_idle</b>—shows that the tunnel has just been created.</li> <li>■ <b>cc_responder_wait_ctl_conn</b>—shows that the tunnel has sent the start control connection response (SCCRP) and is waiting for the start control connection connected (SCCCN) message.</li> <li>■ <b>Cleanup</b>—shows that the tunnel is being cleaned up.</li> <li>■ <b>Closed</b>—shows that the tunnel is being closed.</li> <li>■ <b>Destroyed</b>—shows that the tunnel is being destroyed.</li> <li>■ <b>Established</b>—shows that the tunnel is operational.</li> <li>■ <b>Terminate</b>—shows that the tunnel is being terminated.</li> <li>■ <b>Unknown</b>—shows that the tunnel is not connected to the router.</li> </ul>

**Table 125: jnxL2tpTunnelStatsTable** (continued)

Object	Object Identifier	Description
jnxL2tpTunnelStatsLocalIpAddrType	jnxL2tpTunnelStatsEntry 10	The type of local-end address of the tunnel.
jnxL2tpTunnelStatsLocalIpAddress	jnxL2tpTunnelStatsEntry 11	The local-end address of the tunnel.
jnxL2tpTunnelStatsLocalUdpPort	jnxL2tpTunnelStatsEntry 12	The local-end UDP port number of the tunnel.
jnxL2tpTunnelStatsLocalHostName	jnxL2tpTunnelStatsEntry 13	The local host name of the tunnel.
jnxL2tpTunnelStatsRemoteHostName	jnxL2tpTunnelStatsEntry 14	The host name of the L2TP peer, as discovered during the tunnel establishment phase (via the Host Name AVP). If the tunnel is idle, this object shows the value based on the data from the last time it was connected.
jnxL2tpTunnelMaxSessions	jnxL2tpTunnelStatsEntry 15	The maximum number of sessions configured on the tunnel. Value could be a positive number or zero (unlimited).
jnxL2tpTunnelStatsWindowSize	jnxL2tpTunnelStatsEntry 16	The send window size for the tunnel.
jnxL2tpTunnelStatsHelloInterval	jnxL2tpTunnelStatsEntry 17	The hello interval for the tunnel.
jnxL2tpTunnelStatsCreationTime	jnxL2tpTunnelStatsEntry 18	The time when the tunnel was created.
jnxL2tpTunnelStatsUpTime	jnxL2tpTunnelStatsEntry 19	The time elapsed since the tunnel was established.
jnxL2tpTunnelStatsIdleTime	jnxL2tpTunnelStatsEntry 20	The time elapsed since the last data activity, transmission or reception, on the tunnel.
jnxL2tpTunnelStatsCollectionStart	jnxL2tpTunnelStatsEntry 21	The time at which the statistics gathering started for the tunnel.
jnxL2tpTunnelStatsControlTxPkts	jnxL2tpTunnelStatsEntry 22	The number of control packets transmitted to the tunnel peer.
jnxL2tpTunnelStatsControlTxBytes	jnxL2tpTunnelStatsEntry 23	The number of control bytes transmitted to the tunnel peer.
jnxL2tpTunnelStatsControlRxPkts	jnxL2tpTunnelStatsEntry 24	The number of control packets received on the tunnel.
jnxL2tpTunnelStatsControlRxByte	jnxL2tpTunnelStatsEntry 25	The number of control bytes received from the tunnel peer.
jnxL2tpTunnelStatsDataTxPkts	jnxL2tpTunnelStatsEntry 26	The number of data packets transmitted to the tunnel.
jnxL2tpTunnelStatsDataTxBytes	jnxL2tpTunnelStatsEntry 27	The number of data bytes transmitted to the tunnel peer.
jnxL2tpTunnelStatsDataRxPkts	jnxL2tpTunnelStatsEntry 28	The number of data packets received from the tunnel.
jnxL2tpTunnelStatsDataRxBytes	jnxL2tpTunnelStatsEntry 29	The number of data bytes received from the tunnel peer.
jnxL2tpTunnelStatsErrorTxPkts	jnxL2tpTunnelStatsEntry 30	The number of error transmits packets on the tunnel.
jnxL2tpTunnelStatsErrorRxPkts	jnxL2tpTunnelStatsEntry 31	The number of error receive packets on the tunnel.

## jnxL2tpSessionStatsTable

The jnxL2tpSessionStatsTable, whose object ID is jnxL2tpObjects 4, contains the objects that describe the current status and statistics of a single L2TP tunneled session.

A jnxL2tpSessionStatsEntry represents an L2TP session interface status and has the objects that are listed in Table 126 on page 495.

**Table 126: jnxL2tpSessionStatsTable**

Object	Object Identifier	Description
jnxL2tpSessionStatsLocalTID	jnxL2tpSessionStatsEntry 1	The local tunnel Identifier.
jnxL2tpSessionStatsLocalSID	jnxL2tpSessionStatsEntry 2	The local session Identifier.
jnxL2tpSessionStatsServiceInterface	jnxL2tpSessionStatsEntry 3	The name of the service interface on which this session is being hosted.
jnxL2tpSessionStatsTunnelGroup	jnxL2tpSessionStatsEntry 4	The name of the tunnel group to which this session belongs.
jnxL2tpSessionStatsRemoteSID	jnxL2tpSessionStatsEntry 5	The remote-end assigned session identifier for this session. This value remains zero from the time of starting the session until the time the remote end point responds.
jnxL2tpSessionStatsInterfaceUnit	jnxL2tpSessionStatsEntry 6	The interface unit number that corresponds to the logical service interface on which the session is being hosted.
jnxL2tpSessionStatsEncapType	jnxL2tpSessionStatsEntry 7	The tunnel encapsulation type.
jnxL2tpSessionStatsBundleID	jnxL2tpSessionStatsEntry 8	The ID of the bundle to which the session is linked. This field is valid only for tunnel encapsulation type multilink-ppp.
jnxL2tpSessionStatsStat	jnxL2tpSessionStatsEntry 9	One of the following status messages to show the state of the session at the time of querying: <ul style="list-style-type: none"> <li>■ Established— The session is operational.</li> <li>■ Closed—The session has been closed.</li> <li>■ Destroyed—The session has been destroyed.</li> <li>■ Cleanup—The session has been cleaned up.</li> <li>■ Ins_ic_accept_new—The new session has been accepted.</li> <li>■ Ins_ic_idle—The session has been created but is in idle state.</li> <li>■ Ins_ic_reject_new—A new session has been rejected.</li> <li>■ Ins_ic_wait_connect—The session is waiting for the peer's incoming call connected (ICCN) message.</li> </ul>

**Table 126: jnxL2tpSessionStatsTable** (continued)

Object	Object Identifier	Description
jnxL2tpSessionStatsUserName	jnxL2tpSessionStatsEntry 10	The peer session name on the interface. This is typically the login name of the remote user. This object contains a null string when the user name is unknown to the local tunnel peer.
jnxL2tpSessionStatsMode	jnxL2tpSessionStatsEntry 11	The configured mode value for this session.
jnxL2tpSessionStatsLocalAddrType	jnxL2tpSessionStatsEntry 12	The type of the local-end address of the tunnel that hosts the session.
jnxL2tpSessionStatsLocalAddress	jnxL2tpSessionStatsEntry 13	The local end address of the tunnel that hosts the session.
jnxL2tpSessionStatsLocalUdpPort	jnxL2tpSessionStatsEntry 14	The UDP port of the local end of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteAddrType	jnxL2tpSessionStatsEntry 15	The type of the remote end address of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteAddress	jnxL2tpSessionStatsEntry 16	The remote end address of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteUdpPort	jnxL2tpSessionStatsEntry 17	The UDP port of the remote-end of the tunnel that hosts the session.
jnxL2tpSessionStatsLocalHostName	jnxL2tpSessionStatsEntry 18	The local host name of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteHostName	jnxL2tpSessionStatsEntry 19	The host name as discovered during the tunnel establishment phase (via the Host Name AVP) of the L2TP peer.
jnxL2tpSessionAssignedIpAddrType	jnxL2tpSessionStatsEntry 20	The type of IP address of PPP client being tunneled as obtained from IPCP configuration while establishing the session.
jnxL2tpSessionAssignedIpAddress	jnxL2tpSessionStatsEntry 21	The IP address of the PPP client being tunneled as obtained from IPCP configuration while establishing the session.
jnxL2tpSessionLocalMRU	jnxL2tpSessionStatsEntry 22	The MRU for the local PPP Entity. This value is the MRU that the remote entity uses when sending packets to the session.
jnxL2tpSessionRemoteMRU	jnxL2tpSessionStatsEntry 23	The MRU for the remote PPP Entity. This value is the MRU that the local entity uses when sending packets to the remote PPP client.
jnxL2tpSessionStatsTxSpeed	jnxL2tpSessionStatsEntry 24	The last known transmit baud rate for the session.
jnxL2tpSessionStatsRxSpeed	jnxL2tpSessionStatsEntry 25	The last known receive baud rate for the session.

**Table 126: jnxL2tpSessionStatsTable** (continued)

Object	Object Identifier	Description
jnxL2tpSessionStatsCallBearerType	jnxL2tpSessionStatsEntry 26	The bearer type of this session.
jnxL2tpSessionStatsFramingType	jnxL2tpSessionStatsEntry 27	The framing type of the session.
jnxL2tpSessionStatsLCPRenegotiation	jnxL2tpSessionStatsEntry 28	The ON/OFF state of the LCP renegotiation for the session.
jnxL2tpSessionStatsAuthMethod	jnxL2tpSessionStatsEntry 29	The proxy authentication method employed by the LAC for the session.
jnxL2tpSessionStatsNasIpAddrType	jnxL2tpSessionStatsEntry 30	The type of IP address of the RADIUS network address server to which the accounting records for this session are being sent.
jnxL2tpSessionStatsNasIpAddress	jnxL2tpSessionStatsEntry 31	The IP address of the RADIUS network address server to which the accounting records for the session are being sent.
jnxL2tpSessionStatsNasIpPort	jnxL2tpSessionStatsEntry 32	The port on which RADIUS network address server accounting messages are sent.
jnxL2tpSessionStatsFramedProtocol	jnxL2tpSessionStatsEntry 33	The frame protocol attribute obtained from RADIUS server for the session.
jnxL2tpSessionStatsFramedIpAddrType	jnxL2tpSessionStatsEntry 34	The address to be configured for the user, as provided by the RADIUS server in response to authentication request.
jnxL2tpSessionStatsFramedIpAddress	jnxL2tpSessionStatsEntry 35	The address to be configured for the user, as provided by the RADIUS server in response to the authentication request.
jnxL2tpSessionStatsCallingStationID	jnxL2tpSessionStatsEntry 36	The phone number from which call came in. The RADIUS NAS obtains the phone number that the call came from by using Automatic Number Identification (ANI) or similar technology. It is used only in Access-Request packets.
jnxL2tpSessionStatsCalledStationID	jnxL2tpSessionStatsEntry 37	The phone number to which the user called. The RADIUS NAS obtains the phone number that the user called by using Dialed Number Identification (DNIS) or similar technology. It is used only in Access-Request packets.
jnxL2tpSessionStatsAcctDelayTime	jnxL2tpSessionStatsEntry 38	Duration (in seconds) for which the RADIUS accounting client has been trying to send a record for. This value can be subtracted from the time of arrival on the server to find the approximate time of the event generating this Accounting-Request.
jnxL2tpSessionStatsAcctSessionID	jnxL2tpSessionStatsEntry 39	A unique Accounting ID to help match start and stop records in a log file.

**Table 126: jnxL2tpSessionStatsTable** (continued)

Object	Object Identifier	Description
jnxL2tpSessionStatsAcctMethod	jnxL2tpSessionStatsEntry 40	The accounting method employed for this session.
jnxL2tpSessionStatsAcctSessionTim	jnxL2tpSessionStatsEntry 41	Number of seconds for which the user has received service.
jnxL2tpSessionStatsAcctNasPortType	jnxL2tpSessionStatsEntry 42	The type of the physical port of the NAS that performs accounting for the user.
jnxL2tpSessionStatsAcctTnlClientEndPoint	jnxL2tpSessionStatsEntry 43	This object contains the remote tunnel Identifier of the tunnel that hosts the session.
jnxL2tpSessionStatsAcctTnlServerEndPoint	jnxL2tpSessionStatsEntry 44	The local tunnel Identifier of the tunnel that hosts the session.
jnxL2tpSessionStatsAcctTnlClientAuthID	jnxL2tpSessionStatsEntry 45	The host name of the tunnel that hosts the session as discovered during the tunnel establishment phase (via the Host Name AVP) of the L2TP peer.
jnxL2tpSessionStatsAcctTnlServerAuthID	jnxL2tpSessionStatsEntry 46	The local host name of the tunnel that hosts the session.
jnxL2tpSessionStatsUserProfileName	jnxL2tpSessionStatsEntry 47	The configured access profile name that is being used for the session.
jnxL2tpSessionStatsInterfaceID	jnxL2tpSessionStatsEntry 48	The interface identification (name) for the service interface that bears the session.
jnxL2tpSessionStatsCallSerialNumber	jnxL2tpSessionStatsEntry 49	The serial number assigned to the session.
jnxL2tpSessionStatsCreationTim	jnxL2tpSessionStatsEntry 50	Time when the session was created.
jnxL2tpSessionStatsUpTim	jnxL2tpSessionStatsEntry 51	The time elapsed since the session was established.
jnxL2tpSessionStatsIdleTime	jnxL2tpSessionStatsEntry 52	The time elapsed since the session had any data activity (transmission or reception).
jnxL2tpSessionStatsCollectionStart	jnxL2tpSessionStatsEntry 53	The time at which the statistics gathering started for the session.
jnxL2tpSessionStatsControlTxPkts	jnxL2tpSessionStatsEntry 54	The number of control packets transmitted to the session peer.
jnxL2tpSessionStatsControlTxBytes	jnxL2tpSessionStatsEntry 55	The number of control bytes that were transmitted to the session peer.
jnxL2tpSessionStatsControlRxPkts	jnxL2tpSessionStatsEntry 56	The number of control packets received on the session.
jnxL2tpSessionStatsControlRxBytes	jnxL2tpSessionStatsEntry 57	The number of control bytes received from the session peer.



**Table 126: jnxL2tpSessionStatsTable** (continued)

Object	Object Identifier	Description
jnxL2tpSessionStatsDataTxPkts	jnxL2tpSessionStatsEntry 58	The number of data packets transmitted to the remote session peer.
jnxL2tpSessionStatsDataTxBytes	jnxL2tpSessionStatsEntry 59	The number of data bytes that were transmitted to the session peer.
jnxL2tpSessionStatsDataRxPkts	jnxL2tpSessionStatsEntry 60	The number of data packets received on this session.
jnxL2tpSessionStatsDataRxBytes	jnxL2tpSessionStatsEntry 61	The number of data bytes that were received from the session peer.
jnxL2tpSessionStatsErrorTxPkt	jnxL2tpSessionStatsEntry 62	The number of error transmit packets on the session.
jnxL2tpSessionStatsErrorRxPkts	jnxL2tpSessionStatsEntry 63	The number of error receive packets on the session.

## jnxL2tpMlpppBundleStatsTable

The jnxL2tpMlpppBundleStatsTable, whose object ID is jnxL2tpObjects 5, contains objects that describe the current status and statistics of a single L2TP tunneled multilink PPP bundle.

A jnxL2tpMlpppBundleStatsEntry represents the L2TP MLPPP bundle statistics and has the objects listed in Table 127 on page 499.

**Table 127: jnxL2tpMlpppBundleStatsTable**

Object	Object Identifier	Description
jnxL2tpMlpppBundleStatsBundleID	jnxL2tpMlpppBundleStatsEntry 1	Identifies the session's associated bundle.
jnxL2tpMlpppBundleStatsNumLinks	jnxL2tpMlpppBundleStatsEntry 2	Shows the current number of links that have joined the bundle.
jnxL2tpMlpppBundleStatsEndpoint	jnxL2tpMlpppBundleStatsEntry 3	Shows the username of the MLPPP bundle.
jnxL2tpMlpppBundleStatsInputMrru	jnxL2tpMlpppBundleStatsEntry 4	Shows the maximum packet size that the input interface can process.
jnxL2tpMlpppBundleStatsOutputMrru	jnxL2tpMlpppBundleStatsEntry 5	Shows the maximum packet size that the output interface can process.



## Chapter 47

# Interpreting the Enterprise-Specific Real-Time Performance Monitoring (RPM) MIB

The enterprise-specific Real-Time Performance Monitoring (RPM) Management Information Base (MIB), enables you to access real-time performance-related data over SNMP. Starting with JUNOS Release 8.4, you can access jitter measurements and calculations over SNMP.

The RPM MIB represents a restructuring of the standard Ping MIB and converts the flat structure of the Ping MIB into a hierarchical collection of data. For more information on Ping MIB, see Chapter 25, “Interpreting the Enterprise-Specific Ping MIB.” Similar to the Ping MIB, the RPM MIB too has two groups of tables: the Results group and the History group. The RPM MIB, however, groups its data into separate collection types and measurement sets.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rpm.txt).

This chapter discusses the following topics:

- `jnxRpmResultsSampleTable` on page 501
- `JnxRpmTimestampType` on page 503
- `jnxRpmResultsSummaryTable` on page 503
- `jnxRpmResultsCalculatedTable` on page 504
- `jnxRpmHistorySampleTable` on page 505
- `jnxRpmHistorySummaryTable` on page 506
- `jnxRpmHistoryCalculatedTable` on page 506

## **jnxRpmResultsSampleTable**

---

The `jnxRpmResultsSampleTable` provides you measurements from the latest individual RPM probe samples. Each `jnxRpmResultsSampleEntry` has the objects listed in Table 128 on page 502.



**NOTE:** `jnxRpmResultsSampleTable` does not maintain entries for unsuccessful probes.

---

**Table 128: jnxRpmResultsSampleTable**

Object	Object Identifier	Description
jnxRpmResSampleType	jnxRpmResultsSampleEntry 1	The measurement type for the particular jnxRpmResultsSampleEntry.
jnxRpmResSampleValue	jnxRpmResultsSampleEntry 2	The measurement for the entry.
jnxRpmResSampleTsType	jnxRpmResultsSampleEntry 3	The type of timestamp used to obtain the measurement.
jnxRpmResSampleDate	jnxRpmResultsSampleEntry 4	Date and time at which the measurement was obtained.

- JnxRpmMeasurementType on page 502

## JnxRpmMeasurementType

Different types of measurements can be made for each probe. Table 129 on page 502 lists the measurement types used in jnxRpm.mib.

**Table 129: JnxRpmMeasurementType**

Measurement Type	Description
roundTripTime	The delay between the transmission of a probe and the arrival of its response.
rttJitter	The difference between the current round trip time measurement and the previous one.
rttInterarrivalJitter	<p>An estimate of the statistical variance of a packet's inter-arrival time. Defined in RFC 1889 as:</p> $J = J + ( D(i-1, i)  - J) / 16$ <p>where J is the inter-arrival jitter and D (i-1, i) is the egressJitter measurement.</p>
egress	The delay between the transmission of a probe and its arrival at the destination.
egressJitter	The difference between the current egress delay and the previous measurement.
egressInterarrivalJitter	<p>An estimate of the statistical variance of a packet's inter-arrival time. Defined in RFC 1889 as:</p> $J = J + ( D(i-1, i)  - J) / 16$ <p>where J is the inter-arrival jitter and D (i-1, i) is the egressJitter measurement</p>

**Table 129: JnxRpmMeasurementType** (continued)

Measurement Type	Description
ingress	The delay between the transmission of a probe response and its arrival at the destination.
ingressJitter	The difference between the current ingress delay and the previous measurement.
ingressInterarrivalJitter	<p>An estimate of the statistical variance of a packet's interarrival time. Defined in RFC1889 as:</p> $J = J + ( D(i-1, i) - J ) / 16$ <p>Where J is the interarrival jitter and D (i-1, i) is the current ingressjitter measurement.</p>

Not all types of measurements are performed for every probe. For example, the jitter measurements are available only for those RPM entries that use hardware timestamps on both client and server. Similarly, the ingress and egress measurements are available only for those probe types that measure one-way delays or where hardware timestamps are used (for this, the one-way-hardware timestamp knob must be enabled). However, in the cases discussed above, if the one-way delay is greater than the round-trip time, the corresponding entries are not stored.



**NOTE:** To avoid possible variations in one-way jitter measurements and calculations due to clock synchronization issues, one-way jitter measurements are performed only on samples that are less than 10 seconds apart.

## JnxRpmTimestampType

The following three types of timestamps are used to obtain measurements:

- **software**—Indicates that software-based timestamps are used on both client and server.
- **clientHardware**—Indicates that hardware-based timestamps are used on the client.
- **clientAndServerHardware**—Indicates that hardware-based timestamps are used on the RPM client and the server.

## jnxRpmResultsSummaryTable

The `jnxRpmResultsSummaryTable` provides a summary of the results for each RPM entry (identified by `pingCtlOwnerIndex`/ `pingCtlTestName` in the Ping MIB) and for each data collection maintained by that entry. The RPM feature maintains several different collections of probe data, providing overall summaries as well as detailed calculations for each collection.

The `jnxRpmResultsSummaryTable` maintains the following collection types:

- `currentTest`—The test that is being executed currently.
- `lastCompletedTest`—The most recently completed test.
- `movingAverage`—A list of most recent probes. You can configure the number of probes for this list using `jnxPingCtlMovAvgSize` or the `moving-average-size` CLI command.
- `allTests`—All the probes that were sent. The value gets reset when the 64-bit value storing the square rolls over.

For each collection type, the table provides the following details:

- Number of probes sent
- Number of probes received
- Percentage of probes lost
- Timestamp for the latest sample in the collection

The `jnxRpmResultsSummaryEntry` has the objects listed in Table 130 on page 504.

**Table 130: `jnxRpmResultsSummaryTable`**

Object	Object Identifier	Description
<code>jnxRpmResSumCollection</code>	<code>jnxRpmResultsSummaryEntry 1</code>	The collection of probes to which the <code>jnxRpmResultsSummaryEntry</code> refers.  <b>NOTE:</b> No entries are created for collection types that are not supported or not configured.
<code>jnxRpmResSumSent</code>	<code>jnxRpmResultsSummaryEntry 2</code>	The number of probes sent within the collection.
<code>jnxRpmResSumReceived</code>	<code>jnxRpmResultsSummaryEntry 3</code>	The number of probes received within the collection.
<code>jnxRpmResSumPercentLost</code>	<code>jnxRpmResultsSummaryEntry 4</code>	The percentage of probes that are lost within the collection.
<code>jnxRpmResSumDate</code>	<code>jnxRpmResultsSummaryEntry 5</code>	The timestamp for the most recent probe within the collection.

## **`jnxRpmResultsCalculatedTable`**

The `jnxRpmResultsCalculatedTable` provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported measurement set within that collection of probes.

The `jnxRpmResultsCalculatedEntry` has the objects listed in Table 131 on page 505.

**Table 131: jnxRpmResultsCalculatedTable**

Object	Object Identifier	Description
jnxRpmResCalcSet	jnxRpmResultsCalculatedEntry 1	The measurement set for the particular jnxRpmResultsCalculatedEntry.
jnxRpmResCalcSamples	jnxRpmResultsCalculatedEntry 2	The number of samples used in the calculations.
jnxRpmResCalcMin	jnxRpmResultsCalculatedEntry 3	The minimum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmResCalcMax	jnxRpmResultsCalculatedEntry 4	The maximum (in microseconds) of all the samples in the collection and the measurement set.
jnxRpmResCalcAverage	jnxRpmResultsCalculatedEntry 5	The average (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmResCalcPkToPk	jnxRpmResultsCalculatedEntry 6	The difference (in microseconds) between the minimum and maximum of all the samples in the collection and the measurement set associated with the entry.
jnxRpmResCalcStdDev	jnxRpmResultsCalculatedEntry 7	The standard deviation (in microseconds) calculated over all the samples in the collection and the measurement set associated with the entry.
jnxRpmResCalcSum	jnxRpmResultsCalculatedEntry 8	The sum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.

## jnxRpmHistorySampleTable

The jnxRpmHistorySampleTable provides measurements for each sample stored in the history table of RPM probe entries. In addition to the last completed probe, the table also provides data for a configurable number of most recent probes (all the history tables in this MIB provide the same number of entries as the pingProbeHistoryTable). However, the table does not maintain entries for:

- Unsuccessful probes
- Invalid measurement types

The jnxRpmHistorySampleEntry has the objects listed in Table 132 on page 505.

**Table 132: jnxRpmHistorySampleTable**

Object	Object Identifier	Description
jnxRpmHistSampleType	jnxRpmHistorySampleEntry 1	The measurement type associated with the entry.
jnxRpmHistSampleValue	jnxRpmHistorySampleEntry 2	The measurement for the entry.
jnxRpmHistSampleTsType	jnxRpmHistorySampleEntry 3	The type of timestamp used to obtain the measurement.

## jnxRpmHistorySummaryTable

Similar to the `jnxRpmResultsSummaryTable`, the `jnxRpmHistorySummaryTable` provides you with summary data for each collection of probes within each RPM entry. In addition to summary data for the current probe, the table also provides summary information for a number of the most recent probes. You can configure the number of most recent probes that should be stored in the table.

The `jnxRpmHistorySummaryEntry` has the objects listed in Table 133 on page 506.

**Table 133: jnxRpmHistorySummaryTable**

Object	Object Identifier	Description
<code>jnxRpmHistSumCollection</code>	<code>jnxRpmHistorySummaryEntry 1</code>	The collection of probes associated with the entry.  <b>NOTE:</b> Historical summaries are available only for the current test ( <code>currentTest</code> ).
<code>jnxRpmHistSumSent</code>	<code>jnxRpmHistorySummaryEntry 2</code>	The number of probes sent within the collection.
<code>jnxRpmHistSumReceived</code>	<code>jnxRpmHistorySummaryEntry 3</code>	The number of probes received within the collection.
<code>jnxRpmHistSumPercentLost</code>	<code>jnxRpmHistorySummaryEntry 4</code>	The percentage of probes lost within the collection.

## jnxRpmHistoryCalculatedTable

As with the `jnxRpmResultsCalculatedTable`, the `jnxRpmHistoryCalculatedTable` provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported calculated type within that collection of probes.

In addition to data from the current probe, this table also provides data from a configurable number of the most recent probes.



**NOTE:** The only collection type that is stored in `jnxRpmHistoryCalculatedTable` is the `currentTest`.

Each `jnxRpmHistoryCalculatedEntry` has the objects listed in Table 134 on page 506.

**Table 134: jnxRpmHistoryCalculatedTable**

Object	Object Identifier	Description
<code>jnxRpmHistCalcSet</code>	<code>jnxRpmHistoryCalculatedEntry</code>	The measurement set for the <code>jnxRpmHistoryCalculatedEntry</code> .
<code>jnxRpmHistCalcSamples</code>	<code>jnxRpmHistoryCalculatedEntry 2</code>	The number of samples used in the calculations for this entry.
<code>jnxRpmHistCalcMin</code>	<code>jnxRpmHistoryCalculatedEntry 3</code>	The minimum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.



**Table 134: jnxRpmHistoryCalculatedTable** (continued)

Object	Object Identifier	Description
jnxRpmHistCalcMax	jnxRpmHistoryCalculatedEntry 4	The maximum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcAverage	jnxRpmHistoryCalculatedEntry 5	The average (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcPkToPk	jnxRpmHistoryCalculatedEntry 6	The difference (in microseconds) between the minimum and the maximum of all the samples in the collection and the measurement set associated with the row.
jnxRpmHistCalcStdDev	jnxRpmHistoryCalculatedEntry 7	The standard deviation (in microseconds) calculated over all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcSum	jnxRpmHistoryCalculatedEntry 8	The sum of all the samples in the collection and the measurement set associated with the entry.



## Chapter 48

# Interpreting the Enterprise-Specific Class-of-Service MIB

The enterprise-specific class-of-service (CoS) MIB provides support for monitoring interface output queue statistics per interface and per forwarding class.

The CoS MIB is an object of the `jnxMibs` branch of the enterprise-specific MIB and has an object identifier of `{jnxMIB 15}`. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-cos.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-cos.txt).

This chapter contains the following topic:

- `jnxCosInvQstatTable` on page 509

## `jnxCosInvQstatTable`

This table provides output queue statistics for each interface. Only those physical interfaces that support output queue statistics appear in this table. This table provides the same information as the `jnxCosQstatTable`, but the information is ordered by queue number and interface index, instead of by interface index and queue number.

The entries in the `jnxCosInvQstatTable`, whose object identifier is `{jnxCos 6}`, are represented by `jnxCosInvQstatEntry`, whose object identifier is `{jnxCosIfqStatsTable 1}`, and are listed in Table 135 on page 509.

**Table 135: `jnxCosInvQstatEntry`**

Object	Object Identifier	Description
<code>jnxCosInvQstatQedPkts</code>	<code>jnxCosInvQstatEntry 1</code>	The number of packets queued at the output queue.
<code>jnxCosInvQstatQedPktRate</code>	<code>jnxCosInvQstatEntry 2</code>	The rate (in packets per second) at which packets are queued at the output queue.
<code>jnxCosInvQstatQedBytes</code>	<code>jnxCosInvQstatEntry 3</code>	The number of bytes queued at the output queue.
<code>jnxCosInvQstatQedByteRate</code>	<code>jnxCosInvQstatEntry 4</code>	The rate (in bytes per second) at which bytes are queued at the output queue.
<code>jnxCosInvQstatQedTxedPkts</code>	<code>jnxCosInvQstatEntry 5</code>	The number of packets transmitted on the queue.

**Table 135: jnxCosInvQstatEntry** (continued)

Object	Object Identifier	Description
jnxCosInvQstatQedTxedPktRate	jnxCosInvQstatEntry 6	The packet transmission rate of the output queue (in packets per second).
jnxCosInvQstatQedTxedBytes	jnxCosInvQstatEntry 7	The number of bytes transmitted on the queue.
jnxCosInvQstatQedTxedByteRate	jnxCosInvQstatEntry 8	The byte transmission rate of the output queue (in bytes per second).
jnxCosInvQstatQedTailDropPkts	jnxCosInvQstatEntry 9	The number of packets tail dropped at the output queue.
jnxCosInvQstatQedTailDropPktRate	jnxCosInvQstatEntry 10	The tail drop packet rate (in packets per second) for the queue.
jnxCosInvQstatTotalRedDropPkts	jnxCosInvQstatEntry 11	The number of packets dropped on the interface due to random early detection (RED) at the output.
jnxCosInvQstatTotalRedDropPktRate	jnxCosInvQstatEntry 12	The most-recent estimate of the drop rate (in packets per second) for packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropPkts	jnxCosInvQstatEntry 13	The number of low PLP non-TCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropPktRate	jnxCosInvQstatEntry 14	The rate (in packets per second) at which low PLP non-TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropPkts	jnxCosInvQstatEntry 15	The number of low PLP TCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropPktRate	jnxCosInvQstatEntry 16	The rate (in packets per second) at which low PLP TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropPkts	jnxCosInvQstatEntry 17	The number of high PLP non-TCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropPktRate	jnxCosInvQstatEntry 18	The rate (in packets per second) at which high PLP non-TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropPkts	jnxCosInvQstatEntry 19	The number of high PLP TCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropPktRate	jnxCosInvQstatEntry 20	The rate (in packets per second) at which high PLP TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatTotalRedDropBytes	jnxCosInvQstatEntry 21	The number of bytes dropped on the interface due to RED at the output.
jnxCosInvQstatTotalRedDropByteRate	jnxCosInvQstatEntry 22	The rate (in bytes per second) at which bytes are dropped on the interface due to RED at the output.

**Table 135: jnxCosInvQstatEntry** (continued)

Object	Object Identifier	Description
jnxCosInvQstatLpNonTcpRedDropBytes	jnxCosInvQstatEntry 23	The number of low PLP non-TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropByteRate	jnxCosInvQstatEntry 24	The rate (in bytes per second) at which low PLP non-TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropBytes	jnxCosInvQstatEntry 25	The number of low PLP TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropByteRate	jnxCosInvQstatEntry 26	The rate (in bytes per second) at which low PLP TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropBytes	jnxCosInvQstatEntry 27	The number of high PLP non-TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropByteRate	jnxCosInvQstatEntry 28	The rate (in bytes per second) at which high PLP non-TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropBytes	jnxCosInvQstatEntry 29	The number of high PLP TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropByteRate	jnxCosInvQstatEntry 30	The rate (in bytes per second) at which high PLP TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatLpRedDropPkts	jnxCosInvQstatEntry 31	The number of low PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpRedDropPktRate	jnxCosInvQstatEntry 32	The rate (in packets per second) at which low PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropPkts	jnxCosInvQstatEntry 33	The number of medium-low PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropPktRate	jnxCosInvQstatEntry 34	The rate (in packets per second) at which medium-low PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropPkts	jnxCosInvQstatEntry 35	The number of medium-high PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropPktRate	jnxCosInvQstatEntry 36	The rate (in packets per second) at which medium-high PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHpRedDropPkts	jnxCosInvQstatEntry 37	The number of high PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatHpRedDropPktRate	jnxCosInvQstatEntry 38	The rate (in packets per second) at which high PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatLpRedDropBytes	jnxCosInvQstatEntry 39	The number of low PLP bytes dropped on the interface due to RED at the output.

**Table 135: jnxCosInvQstatEntry** (continued)

Object	Object Identifier	Description
jnxCosInvQstatLpRedDropByteRate	jnxCosInvQstatEntry 40	The rate (in bytes per second) at which low PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropBytes	jnxCosInvQstatEntry 41	The number of medium-low PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropByteRate	jnxCosInvQstatEntry 42	The rate (in bytes per second) at which medium-low PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropBytes	jnxCosInvQstatEntry 43	The number of medium-high PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropByteRate	jnxCosInvQstatEntry 44	The rate (in bytes per second) at which medium-high PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHpRedDropBytes	jnxCosInvQstatEntry 45	The number of high PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHpRedDropByteRate	jnxCosInvQstatEntry 46	The rate (in bytes per second) at which high PLP bytes are dropped on the interface due to RED at the output.

## Chapter 49

# Interpreting the Enterprise-Specific IP Forward MIB

The enterprise-specific IP Forward MIB, whose object identifier is {*jnxMibs* 38}, extends the *ipCidrRouteTable* in the IP Forwarding Table MIB (as defined in RFC 2096) to include a tunnel name when the next hop is through an RSVP-signaled LSP.

This MIB adds an *jnxIpCidrRouteTunnelName* attribute to the *ipCidrRouteTable*. The attribute exists for each entry in the *ipCidrRouteTable*. (One entry in the *ipCidrRouteTable* represents each route in *inet.0*). If the route's next hop is an RSVP-signaled MPLS LSP, the new attribute contains the LSP name. If the route's next hop is not an RSVP-signaled MPLS LSP, the new attribute is defined as null.

The attribute's name is *jnxIpCidrRouteTunnelName*. Its OID is .1.3.6.1.4.1.2636.3.38.1.1.1. As with any SNMP attribute, an index is appended to the OID to form the instance identifier. Because this attribute augments the *ipCidrRouteTable*, the index is identical to that used in the *ipCidrRouteTable*. The index is formed by concatenating destination address, subnet mask, tos byte, and next hop.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipforward.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipforward.txt).

This chapter contains the following topic:

- *jnxIpCidrRouteTable* on page 513

## ***jnxIpCidrRouteTable***

---

The *jnxIpCidrRouteTable*, whose object identifier is {*jnxIpForwardMIB* 1}, extends the *ipCidrRouteTable* with additional data.

## ***jnxIpCidrRouteEntry***

*jnxIpCidrRouteEntry*, whose object identifier is {*jnxIpCidrRouteTable* 1}, has one object, which is listed in Table 136 on page 514.

**Table 136: jnxlpCidrRouteTable**

Object	Object Identifier	Description
jnxlpCidrRouteTunnelName	nxlpCidrRouteEntry 1	The canonical name assigned to the tunnel. The router forwards traffic bound for the destination through this tunnel.



## Chapter 50

# Interpreting the Enterprise-Specific ATM Class-of-Service MIB

The enterprise-specific ATM Class-of-Service (CoS) Management Information Base (MIB) provides information on the ATM CoS infrastructure.

The Juniper Networks enterprise-specific ATM CoS MIB uses the following objects and definitions as per the RFCs and MIBs:

- `ifIndex` (*RFC 2233, IF MIB*)
- `atmVclVpi` and `atmVclVci` (*RFC 2515, ATM MIB*)
- `jnxMibs` (*Juniper Networks enterprise-specific SMI MIB*) and `jnxCoSFclf` (*Juniper Networks enterprise-specific CoS MIB*)

For a downloadable version of the MIB, see  
[www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-atm-cos.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-atm-cos.txt)

This chapter contains the following topics:

- `jnxCosAtmVcTable` on page 515
- `jnxCosAtmVcScTable` on page 516
- `jnxCosAtmVcQstatsTable` on page 518
- `jnxCosAtmTrunkTable` on page 518

## **jnxCosAtmVcTable**

---

The `jnxCosAtmVcTable`, whose object identifier is `{jnxAtmCos 1}`, contains information about virtual circuits (VC) that have CoS configured.

Each `jnxCosAtmVcEntry` (object identifier is `jnxCosAtmVcTable 1`) is indexed using `ifIndex`, `atmVclVpi`, and `atmVclVci`, and contains the `jnxCosAtmVcCosMode` object. The `jnxCosAtmVcCosMode` object represents the mode of CoS queue priority for the VC.

There are two modes, **strict** mode and **alternate** mode represented by integer values 0 and 1 respectively:

- **Strict** mode (represented by integer value 0): One of the four queues has high priority, and is always serviced before the other three queues. The remaining queues are serviced in a round robin fashion.

- **Alternate mode** (represented by integer value 1): Even though one of the four queues has high priority, the servicing of the queues alternates between the high priority queue and the other queues.

## jnxCosAtmVcScTable

The `jnxCosAtmVcScTable`, whose object identifier is `jnxAtmCos 2`, contains ATM-scheduler configuration parameters for each forwarding class within a specified VC.



**NOTE:** The schedulers referred in this MIB are specific to an ATM interface, and are different from the typical schedulers specified by the Class of Service configuration CLI.

Each `jnxCosAtmVcScEntry` (object identifier is `jnxCosAtmVcScTable 1`) represents ATM-scheduler configuration parameters per forwarding class and per VC, and is indexed using `ifIndex`, `atmVclVpi`, and `atmVclVci`. Table 137 on page 516 lists the ATM scheduler parameters as represented by a `jnxCosAtmVcScEntry`.

**Table 137: jnxCosAtmVcScTable**

Object	Object ID	Description
<code>jnxCosAtmVcScPriority</code>	<code>jnxCosAtmVcScEntry 1</code>	Represents the ATM scheduler priority for the queue associated with the specified forwarding class within the VC.
<code>jnxCosAtmVcScTxWeightType</code>	<code>jnxCosAtmVcScEntry 2</code>	Represents the ATM scheduler transmit-weight-type for the queue associated with the specified forwarding class inside the VC. The transmit-weight-type is represented either as the number of cells or as a percentage of the queue size.
<code>jnxCosAtmVcScTxWeight</code>	<code>jnxCosAtmVcScEntry 3</code>	Represents the transmit weight of the ATM scheduler for the queue associated with the specified forwarding class and the VC. <code>jnxCosAtmVcScTxWeight</code> is expressed either as the number of cells or as a percentage of the total VC bandwidth. The value of <code>jnxCosAtmVcScTxWeightType</code> determines the unit used.

**Table 137: jnxCosAtmVcScTable** (continued)

Object	Object ID	Description
jnxCosAtmVcScDpType	jnxCosAtmVcScEntry 4	Shows the type of RED drop profile configured for the specified forwarding class within the VC. A scheduler can specify either linear or constant drop profile. A constant type drop profile (also known as EPD) specifies that all the cells should be dropped when the number of queued cells exceeds a threshold. A linear type drop profile specifies that only a percentage of cells be dropped based on the number of queued cells at any time.
jnxCosAtmVcScLrdpQueueDepth	jnxCosAtmVcScEntry 5	Represents the maximum queue size in cells, as specified by the linear RED drop profile associated with the specified forwarding class within the VC. This object is valid only when the value of the object jnxCosAtmVcScDpType is linearRed(0).
jnxCosAtmVcScLrdpLowPlpThresh	jnxCosAtmVcScEntry 6	Represents the threshold percentage of fill-level beyond which the low Packet Loss Priority (PLP) packets belonging to the specified forwarding class within the VC are randomly dropped. This value is specified by the linear RED drop profile configuration. This object is valid only when the object jnxCosAtmVcScDpType is set to linearRed(0).
jnxCosAtmVcScLrdpHighPlpThresh	jnxCosAtmVcScEntry 7	Represents the threshold percentage of the fill level beyond which high PLP packets belonging to the specified forwarding class within the VC are randomly dropped. This value is specified by the linear RED drop profile configuration. This object is valid only when the object jnxCosAtmVcScDpType is set to linearRed(0).
jnxCosAtmVcScEpdThreshold	jnxCosAtmVcScEntry 8	Shows the EPD drop threshold configured for the specified forwarding class within the VC. When the number of cells queued exceeds the value specified for this object, all the cells in the queue are dropped. This object is valid only when the jnxCosAtmVcScDpType object is set to epd(1).

## jnxCosAtmVcQstatsTable

The `jnxCosAtmVcQstatsTable` (object identifier is `jnxAtmCos 3`) contains queue statistics for VCs and forwarding classes. Each `jnxCosAtmVcQstatsEntry` in the `jnxCosAtmVcQstatsTable` contains the queue status information for a particular forwarding class and VC. The `jnxCosAtmVcQstatsEntry` object uses `ifIndex`, `atmVclVpi`, `atmVclVci`, and `jnxCosFclId` for indexing.

Each `jnxCosAtmVcQstatsEntry` contains the objects listed in Table 138 on page 518.

**Table 138: jnxCosAtmVcQstatsTable**

Object	Object ID	Description
<code>jnxCosAtmVcQstatsOutPackets</code>	<code>jnxCosAtmVcQstatsEntry 1</code>	Represents the number of packets belonging to a particular forwarding class that is transmitted on a specific VC.
<code>jnxCosAtmVcQstatsOutBytes</code>	<code>jnxCosAtmVcQstatsEntry 2</code>	Represents the number of bytes of a particular forwarding class that are transmitted on a specific VC.
<code>jnxCosAtmVcQstatsOutRedDropPkts</code>	<code>jnxCosAtmVcQstatsEntry 3</code>	Represents the number of RED-dropped outgoing packets of a particular forwarding class that are transmitted on a specific VC.
<code>jnxCosAtmVcQstatsOutNonRedDrops</code>	<code>jnxCosAtmVcQstatsEntry 4</code>	Represents the number of outgoing packets, of a particular forwarding class and transmitted on a specific VC, that are dropped because of errors in packets.
<code>jnxCosAtmVcQstatsOutLpBytes</code>	<code>jnxCosAtmVcQstatsEntry 5</code>	Represents the number of low PLP (PLP0) bytes transmitted.
<code>jnxCosAtmVcQstatsOutLpPkts</code>	<code>jnxCosAtmVcQstatsEntry 6</code>	Represents the number of low PLP (PLP0) packets that are transmitted.
<code>jnxCosAtmVcQstatsOutLpDropBytes</code>	<code>jnxCosAtmVcQstatsEntry 7</code>	Represents the number of low PLP (PLP0) bytes dropped at the output queue.
<code>jnxCosAtmVcQstatsOutHpDropBytes</code>	<code>jnxCosAtmVcQstatsEntry 8</code>	Represents the number of high PLP (PLP1) bytes dropped at the output queue.
<code>jnxCosAtmVcQstatsOutLpDropPkts</code>	<code>jnxCosAtmVcQstatsEntry 9</code>	Represents the number of low PLP (PLP0) packets dropped at the output queue.
<code>jnxCosAtmVcQstatsOutHpDropPkts</code>	<code>jnxCosAtmVcQstatsEntry 10</code>	Represents the number of high PLP (PLP1) packets dropped at the output queue.

## jnxCosAtmTrunkTable

The `jnxCosAtmTrunkTable` (object identifier is `jnxAtmCos 4`) contains statistics and configuration information related to ATM Trunk CoS interface.

The `jnxCosAtmTrunkEntry` (object identifier is `jnxCosAtmTrunkTable 1`) object uses `ifIndex` and `jnxCosFclId`, and contains the objects listed in Table 139 on page 519.

**Table 139: jnxCosAtmTrunkTable**

Object	Object ID	Description
jnxCosAtmTrunkMode	jnxCosAtmTrunkEntry 1	Represents the mode of CoS queue priority for the trunk: <ul style="list-style-type: none"> <li>■ <b>Strict</b> mode (represented by integer value 0): One of the four queues has high priority, and is always serviced before the other three queues. The remaining queues are serviced in a round-robin fashion.</li> <li>■ <b>Alternate</b> mode (represented by integer value 1): Even though one of the four queues has high priority, the servicing of the queues alternates between the high priority queue and the other queues.</li> </ul>
jnxCosAtmTrunkScPriority	jnxCosAtmTrunkEntry 2	Represents the ATM scheduler priority for the queue associated with a particular forwarding class within the trunk.
jnxCosAtmTrunkScTxWeightType	jnxCosAtmTrunkEntry 3	Represents the ATM scheduler transmit weight type for the queue associated with a particular forwarding class inside the trunk. The weight type can be expressed either as the number of cells or as a percentage of the queue size.
jnxCosAtmTrunkScTxWeight	jnxCosAtmTrunkEntry 4	Represents the transmit weight for the queue. The transmit weight can be expressed either as the number of cells or as a percentage of the total trunk bandwidth. The unit is determined by the value set for jnxCosAtmTrunkScTxWeightType.
jnxCosAtmTrunkQaType	jnxCosAtmTrunkEntry 5	Represents the ATM queue admission type used for the specified trunk. Available values for this object are: <b>red</b> (1), <b>singleEpd</b> (2), and <b>dualEpd</b> (3)
jnxCosAtmTrunkEpdThresholdPlp0	jnxCosAtmTrunkEntry 6	Represents the threshold value beyond which all PLP0 cells get dropped. This object has a valid value only when the value for jnxCosAtmTrunkQaType is set to <b>singleEpd</b> or <b>dualEpd</b> .
jnxCosAtmTrunkEpdThresholdPlp1	jnxCosAtmTrunkEntry 7	Represents the threshold value beyond which all PLP1 cells get dropped. This object has a valid value only when the jnxCosAtmTrunkQaType object is set to <b>dualEpd</b> .
jnxCosAtmTrunkQstatsOutPackets	jnxCosAtmTrunkEntry 8	Represents the number of packets that belong to a particular forwarding class, and are transmitted on the specific trunk.
jnxCosAtmTrunkQstatsOutBytes	jnxCosAtmTrunkEntry 9	Represents the number of bytes that belong to a particular forwarding class, and are transmitted on the specific trunk.
jnxCosAtmTrunkQstatsOutDrops	jnxCosAtmTrunkEntry 10	Represents the number of outgoing packets on the trunk that are dropped.
jnxCosAtmTrunkQstatsOutLpBytes	jnxCosAtmTrunkEntry 11	Represents the number of low PLP (PLP0) bytes that are transmitted on the trunk.
jnxCosAtmTrunkQstatsOutLpPkts	jnxCosAtmTrunkEntry 12	Represents the number of low PLP (PLP0) packets that are transmitted on the trunk.
jnxCosAtmTrunkQstatsOutLpDropBytes	jnxCosAtmTrunkEntry 13	Represents the number of low PLP (PLP0) bytes dropped at the output queue.

**Table 139: jnxCosAtmTrunkTable** (continued)

Object	Object ID	Description
jnxCosAtmTrunkQstatsOutHpDropBytes	jnxCosAtmTrunkEntry 14	Represents the number of high PLP (PLP1) bytes that are dropped at the output queue.
jnxCosAtmTrunkQstatsOutLpDropPkts	jnxCosAtmTrunkEntry 15	Represents the number of low PLP (PLP0) packets that are dropped at the output queue.
jnxCosAtmTrunkQstatsOutHpDropPkts	jnxCosAtmTrunkEntry 16	Represents the number of high PLP (PLP1) packets dropped at the output queue.
jnxCosAtmTrunkQstatsOutHpBytes	jnxCosAtmTrunkEntry 17	Represents the number of high PLP (PLP1) bytes that are transmitted on the trunk.
jnxCosAtmTrunkQstatsOutHpPkts	jnxCosAtmTrunkEntry 18	Represents the number of high PLP (PLP1) packets that are transmitted on the trunk.

## Chapter 51

# Interpreting the Enterprise-Specific Firewall MIB

The enterprise-specific Firewall MIB, whose object identifier is {jnxMibs 5}, contains information about firewall filters and policies.

Firewall MIB contains 2 tables, jnxFirewallsTable and jnxFirewallCounterTable.

The jnxFirewallsTable does not support the following conditions:

- Counter and filter names that have more than 24 characters.
- Duplicate counter names, even if the counter types are different.

Because of the preceding limitations, the jnxFirewallsTable has been deprecated and replaced with jnxFirewallCounterTable. However, for backward compatibility, the jnxFirewallsTable is retained in the Firewall MIB.

For a downloadable version of the MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-firewall.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-firewall.txt)

This chapter contains the following topics:

- jnxFirewallsTable on page 521
- jnxFirewallCounterTable on page 522

## jnxFirewallsTable

The deprecated jnxFirewallsTable contains jnxFirewallsEntry, whose object ID is {jnxFirewallsTable 1}. Each jnxFirewallsEntry contains the objects listed in Table 140 on page 521.

**Table 140: jnxFirewallsEntry**

Object	Object ID	Description
jnxFWFilter	jnxFirewallsEntry 1	The name of the firewall filter. This object does not support names that have more than 24 characters.
jnxFWCounter	jnxFirewallsEntry 2	The name of the counter or the policer. This name is specific within the firewall filter.

**Table 140: jnxFirewallsEntry (continued)**

Object	Object ID	Description
jnxFWType	jnxFirewallsEntry 3	The type of the jnxFWCounter object. The value of jnxFWType can be 1 (other), 2 (counter), or 3 (policer).
jnxFWPackets	jnxFirewallsEntry 4	The number of packets that are associated with the specified counter or policer.
jnxFWBytes	jnxFirewallsEntry 5	The number of bytes that are associated with the counter. For policers, the value of jnxFWBytes is always zero because the policers do not count the number of bytes.

## jnxFirewallCounterTable

The jnxFirewallCounterTable, whose object identifier is jnxFirewalls 2, replaces the deprecated jnxFirewallsTable. Each JnxFirewallCounterEntry contains the objects listed in Table 141 on page 522.

**Table 141: JnxFirewallCounterEntry**

Object	Object ID	Description
jnxFWCounterFilterName	jnxFirewallCounterEntry 1	The name of the firewall filter. The name can have up to 127 characters.
jnxFWCounterName	jnxFirewallCounterEntry 2	The name of the counter or the policer. The name can have up to 127 characters.
jnxFWCounterType	jnxFirewallCounterEntry 3	The type of the jnxFWCounterName object. The value of jnxFWType can be 1 (other), 2 (counter), or 3 (policer).
jnxFWCounterPacketCount	jnxFirewallCounterEntry 4	The number of packets that are associated with the specified counter or policer.
jnxFWCounterByteCount	jnxFirewallCounterEntry 5	The number of bytes that are associated with the counter. For policers, the value of jnxFWCounterByteCount is always zero because the policers do not count the number of bytes.
jnxFWCounterDisplayFilterName	jnxFirewallCounterEntry 6	The name of the firewall filter. The name can have up to 127 characters.
jnxFWCounterDisplayName	jnxFirewallCounterEntry 7	The name of the counter or the policer.
jnxFWCounterDisplayType	jnxFirewallCounterEntry 8	The type of the jnxFWCounterName object. The value of jnxFWType can be 1 (other), 2 (counter), or 3 (policer).



## Chapter 52

# Interpreting the Enterprise-Specific ATM MIB

The enterprise-specific ATM MIB, whose object identifier is {jnxMibs 10}, extends the standard *ATM MIB, RFC 1695*, and contains information about ATM interfaces and VCs.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-atm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-atm.txt)

This chapter contains the following topics:

- jnxAtmIfTable on page 523
- jnxAtmVCTable on page 525
- jnxAtmVpTable on page 528
- jnxAtmTrunkTable on page 530

## jnxAtmIfTable

---

The **jnxAtmIfTable** augments the **atmInterfaceConfTable** in the standard ATM MIB. The object identifier for **jnxAtmIfTable** is {jnxAtm 1}. Each **jnxAtmIfEntry** in the **jnxAtmIfTable** contains the configuration and statistic information for a particular ATM port. The **jnxAtmIfEntry**, whose object identifier is {jnxAtmIfTable 1}, is an extension of the **atmInterfaceConfEntry** in the standard ATM MIB.

Table 142 on page 524 lists the objects associated with the **jnxAtmIfEntry**.

**Table 142: jnxAtmIfTable**

Object	Object ID	Description
jnxAtmIfPortType	jnxAtmIfEntry 1	Represents the type of the physical port. This object uses the following integer values to denote the port type: <ul style="list-style-type: none"> <li>■ 1 (other)</li> <li>■ 2 (oc3)</li> <li>■ 3 (oc1)</li> <li>■ 4 (t3)</li> <li>■ 5 (e3)</li> <li>■ 6 (oc48)</li> </ul>
jnxAtmIfEncaps	jnxAtmIfEntry 2	Represents the type of ATM physical link layer encapsulation. This object uses the following integer values to denote the type of encapsulation: <ul style="list-style-type: none"> <li>■ 1 (other)</li> <li>■ 2 (atmPvc)</li> <li>■ 3 (atmCccCellRelay)</li> </ul>
jnxAtmIfLpBackInfo	jnxAtmIfEntry 3	Represents the loopback configuration and type. This object uses the following integer values to denote the loopback configuration type: <ul style="list-style-type: none"> <li>■ 1 (noLoopBack)</li> <li>■ 2 (localLoopBack)</li> <li>■ 3 (remoteLoopBack)</li> </ul>
jnxAtmIfScrambleEnable	jnxAtmIfEntry 4	Indicates whether scrambling is enabled (object value 1) or disabled (object value 2). Default value for this object is 2, disabled.
jnxAtmIfTxCellCount	jnxAtmIfEntry 5	Indicates the number of ATM cells, including the idle cells, transmitted by the interface.
jnxAtmIfRxCellCount	jnxAtmIfEntry 6	Indicates the number of ATM cells, excluding the idle cells, received by the interface.
jnxAtmIfTxIdleCellCount	jnxAtmIfEntry 7	Indicates the number of idle cells transmitted by the interface. When the interface does not have anything to send, it sends idle cells to fill the time slot.
jnxAtmIfUncorrHCSErrs	jnxAtmIfEntry 8	Indicates the number of uncorrectable cell Header Check Sequence (HCS) errors that occurred on the interface.
jnxAtmIfCorrHCSErrs	jnxAtmIfEntry 9	Indicates the number of correctable cell Header Check Sequence (HCS) errors.

**Table 142: jnxAtmIfTable** (continued)

Object	Object ID	Description
jnxAtmIfTxCellFIFOOverRuns	jnxAtmIfEntry 10	Indicates the number of overruns in the Transmit FIFO.
jnxAtmIfRxCellFIFOOverRuns	jnxAtmIfEntry 11	Indicates the number of overruns in the Receive FIFO.
jnxAtmIfRxCellFIFOUnderRuns	jnxAtmIfEntry 12	Indicates the number of underruns in the Receive FIFO.
jnxAtmIfInInvalidVCCells	jnxAtmIfEntry 13	Indicates the number of cells that are arrived for a non-existent VC.
jnxAtmIfInNoBufferOAMCells	jnxAtmIfEntry 14	Indicates the number of incoming OAM cells or raw cells that are dropped because of lack of buffer to handle them.
jnxAtmIfInNoBufDropPkts	jnxAtmIfEntry 15	Indicates the number of AAL5 packets that are dropped because of lack of buffer to handle them.
jnxAtmIfOutVCQueueDrops	jnxAtmIfEntry 16	Indicates the number of packets dropped because of queue limits on each VC.
jnxAtmIfInBadCrcs	jnxAtmIfEntry 17	Indicates the total number of incoming CRC errors.
jnxAtmIfInLenErrPkts	jnxAtmIfEntry 18	Indicates the number of AAL5 packets that were dropped because of incorrect length.
jnxAtmIfInTimeoutPkts	jnxAtmIfEntry 19	Indicates the number of AAL5 packets that were dropped because of reassembly timeout.
jnxAtmIfL2CircuitMode	jnxAtmIfEntry 20	Indicates the layer 2 circuit mode of the ATM interface (on an ATMII PIC). This object uses the following integer values to denote the circuit modes: <ul style="list-style-type: none"> <li>■ 1 (not applicable)</li> <li>■ 2 (none)</li> <li>■ 3 (aal5)</li> <li>■ 4 (cell)</li> <li>■ 5 (uniTrunk)</li> <li>■ 6 (nniTrunk)</li> </ul>

## jnxAtmVCTable

The `jnxAtmVCTable`, whose object identifier is `jnxAtm 2`, extends the `atmVcITable` in the standard ATM MIB, and contains information on P2P, P2MP, and multicast virtual circuit entries.

Each `jnxAtmVCEntry`, whose object identifier is `jnxAtmVCTable 1`, in the `jnxAtmVCTable` contains the objects listed in Table 143 on page 526.

However, the `jnxAtmVCEntry` does not store any statistic for multicast VCs. A value of zero indicates this. Similarly, a value of `0.0.0.0` indicates that the multipoint destination IP address for a P2P VC is invalid. If `atmCccCellRelay` is set as the encapsulation type for the logical interface (to which the VC belongs), only the generic VC info is valid. And, if OAM is disabled (indicated by zero value for `jnxAtmVCFlags`), then all the OAM config and f5 statistics information is invalid.

**Table 143: jnxAtmVCTable**

Object	Object ID	Description
<code>jnxAtmVConnType</code>	<code>jnxAtmVCEntry 1</code>	<p>Indicates the type of connection. This object uses the following integer values to denote the connection types:</p> <ul style="list-style-type: none"> <li>■ 1 (other; unknown connection type or a connection type that is not one of the other connection types explicitly represented)</li> <li>■ 2 (p2p)</li> <li>■ 3 (p2mp; point to multipoint)</li> <li>■ 4 (multicast)</li> </ul>
<code>jnxAtmVCEncapsulation</code>	<code>jnxAtmVCEntry 2</code>	<p>Indicates the ATM encapsulation type associated with the VC. This object uses the following integer values to denote the encapsulation types:</p> <ul style="list-style-type: none"> <li>■ 1 other</li> <li>■ 2 <code>atmCccCellRelay</code> (ATM cell relay for CCC)</li> <li>■ 3 <code>atmCccVcMux</code> (ATM VC for CCC)</li> <li>■ 4 <code>atmCiscoNlpid</code> (Cisco-compatible ATM NLPID encapsulation)</li> <li>■ 5 <code>atmNlpid</code> (ATM NLPID encapsulation)</li> <li>■ 6 <code>atmSnap</code> (ATM LLC/SNAP encapsulation)</li> <li>■ 7 <code>atmVcMux</code> (ATM VC multiplexing)</li> <li>■ 8 <code>atmTccVcmux</code> (Translational Cross Connection (TCC) over ATM VC MUX encapsulation)</li> <li>■ 9 <code>atmTccSnap</code> (TCC over ATM LLC/SNAP encapsulation)</li> </ul>

**Table 143: jnxAtmVCTable** (continued)

Object	Object ID	Description
jnxAtmVCMpDestIPv4Addr	jnxAtmVCEntry 3	Represents the multipoint destination IPv4 address for a P2MP connection. This object returns an all zero address in the following cases: <ul style="list-style-type: none"> <li>■ When the connection type is not P2MP.</li> <li>■ When the multipoint destination address type is IPv6.</li> </ul>
jnxAtmVCMpDestIPv6Addr	jnxAtmVCEntry 4	Represents the multipoint destination IPv6 address for a P2MP connection. This object returns an all zero address in the following cases: <ul style="list-style-type: none"> <li>■ When the connection type is not P2MP.</li> <li>■ When the multipoint destination address type is IPv4.</li> </ul>
jnxAtmVCFlags	jnxAtmVCEntry 5	Contains the flags related to the VC.
jnxAtmVCTotalDownTime	jnxAtmVCEntry 6	Shows the total downtime for the VC after the last reboot of the system.
jnxAtmVCInBytes	jnxAtmVCEntry 7	Represents the number of bytes received on the VC.
jnxAtmVCOutBytes	jnxAtmVCEntry 8	Represents the number of bytes transmitted from the VC.
jnxAtmVCInPkts	jnxAtmVCEntry 9	Represents the number of packets received on the VC.
jnxAtmVCOutPkts	jnxAtmVCEntry 10	Represents the number of packets transmitted from the VC.
jnxAtmVCTailQueuePktDrops	jnxAtmVCEntry 11	Represents the number of packets that were dropped because of bandwidth constraints.
jnxAtmVCOAMPeriod	jnxAtmVCEntry 12	Shows the frequency at which the F5 cells are transmitted to check the status of the VC.
jnxAtmVCOAMUpCellCount	jnxAtmVCEntry 13	Shows the minimum number of loopback cells that are required to confirm that a VC is up.
jnxAtmVCOAMDownCellCount	jnxAtmVCEntry 14	Shows the minimum number of loopback cells that are required to confirm that a VC is down. <p><b>NOTE:</b> This object returns a zero value if OAM is not enabled.</p>

**Table 143: jnxAtmVCTable** (continued)

Object	Object ID	Description
jnxAtmVCInOAMF5LoopCells	jnxAtmVCEntry 15	Shows the number of OAM F5 loopback cells received on a VC.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5LoopCells	jnxAtmVCEntry 16	Shows the number of OAM F5 loopback cells transmitted from a VC.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.
jnxAtmVCInOAMF5RDICells	jnxAtmVCEntry 17	Shows the number of OAM F5 cells that are received with RDI (Remote Defect Indication) bit set.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5RDICells	jnxAtmVCEntry 18	Shows the number of OAM F5 cells that are transmitted with RDI (Remote Defect Indication) bit set.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.
jnxAtmVCInOAMF5AISCells	jnxAtmVCEntry 19	Shows the number of OAM F5 cells that are received with AIS (Alarm Indication Signal) bit set.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5AISCells	jnxAtmVCEntry 20	Shows the number of OAM F5 cells that are transmitted with AIS bit set.  <b>NOTE:</b> This object returns a zero value if OAM is not enabled.

## jnxAtmVpTable

The `jnxAtmVpTable` extends the `atmVpITable` defined in *RFC 2515, ATM MIB*, and contains additional information on ATM virtual paths (VP).

The `jnxAtmVpTable`, whose object identifier is `jnxAtm 3`, contains `jnxAtmVpEntry`. Each `jnxAtmVpEntry`, whose object ID is `jnxAtmVpTable 1`, contains the objects listed in Table 144 on page 529.

**Table 144: jnxAtmVpTable**

Object	Object ID	Description
jnxAtmVpEntry	jnxAtmVpTable 1	<p>Represents configuration status and statistics information related to an ATM VP.</p> <p>However, traffic stats are available per VP tunnel only If shaping is configured on the VP. You can use the <b>jnxAtmVpFlags</b> to determine whether shaping is enabled.</p> <p>Similarly, the values for OAM config and OAM stat objects are invalid (default value: 0) if no OAM is configured. You can use the <b>jnxAtmVpFlags</b> to determine whether OAM is configured.</p> <p><b>NOTE:</b> For an ATM-1 VP, the only valid object is <b>jnxAtmVpFlags</b>.</p>
jnxAtmVpFlags	jnxAtmVpEntry 1	<p>Represents the flags associated with the VP. This object uses the following values:</p> <ul style="list-style-type: none"> <li>■ 0 active</li> <li>■ 1 down</li> <li>■ 2 oamEnabled</li> <li>■ 3 shapingEnabled</li> <li>■ 4 passiveOam</li> </ul>
jnxAtmVpTotalDownTime	jnxAtmVpEntry 2	Represents the total downtime for the VP since the last reboot of the system.
jnxAtmVpOamPeriod	jnxAtmVpEntry 3	<p>Indicates the frequency at which the OAM F4 cells are transmitted to find out the status of the VP.</p> <p>This object returns a value of zero if OAM is not enabled for the VP.</p>
jnxAtmVpOamUpCellCount	jnxAtmVpEntry 4	Indicates the minimum number of consecutive loopback cells required to confirm that a VP is up.
jnxAtmVpOamDownCellCount	jnxAtmVpEntry 5	Indicates the minimum number of consecutive loopback cells required to confirm that a VP is down.
jnxAtmVpInBytes	jnxAtmVpEntry 6	Indicates the number of bytes received on the VP.
jnxAtmVpOutBytes	jnxAtmVpEntry 7	Indicates the number of bytes sent out of the VP.
jnxAtmVpInPkts	jnxAtmVpEntry 8	Indicates the number of packets received on the VP.
jnxAtmVpOutPkts	jnxAtmVpEntry 9	Indicates the number of packets sent out on the VP.
jnxAtmVpInOamF4Cells	jnxAtmVpEntry 10	Indicates the number of OAM F4 cells received on the VP.
jnxAtmVpOutOamF4Cells	jnxAtmVpEntry 11	Indicates the number of OAM F4 cells transmitted on the VP.
jnxAtmVpInOamF4LoopCells	jnxAtmVpEntry 12	Indicates the number of OAM F4 loopback cells received on the VP.
jnxAtmVpOutOamF4LoopCells	jnxAtmVpEntry 13	Indicates the number of OAM F4 cells transmitted on the VP.
jnxAtmVpInOamF4RdiCells	jnxAtmVpEntry 14	Indicates the number of OAM F4 RDI cells received on the VP.

**Table 144: jnxAtmVpTable** (continued)

Object	Object ID	Description
jnxAtmVpOutOamF4RdiCells	jnxAtmVpEntry 15	Indicates the number of OAM F4 RDI cells transmitted on the VP.
jnxAtmVpInOamF4AisCells	jnxAtmVpEntry 16	Indicates the number of OAM F4 AIS cells received on the VP.

## jnxAtmTrunkTable

The `jnxAtmTrunkTable`, whose object identifier is `jnxAtm 4`, contains information related to ATM trunks. Each `JnxAtmTrunkEntry` in `jnxAtmTrunkTable` contains the objects listed in Table 145 on page 530.



**NOTE:** If the encapsulation type for the logical interface to which the trunk belongs is `atmCccCellRelay`, only the generic trunk information (`jnxAtmTrunkConnType`, `jnxAtmTrunkEncapsulation`, `nxAtmTrunkFlags`, and `jnxAtmTrunkTotalDownTime`) is valid.

**Table 145: jnxAtmTrunkTable**

Object	Object ID	Description
jnxAtmTrunkId	jnxAtmTrunkEntry 1	Represents the identifier of the ATM trunk.
jnxAtmTrunkConnType	jnxAtmTrunkEntry 2	Indicates the type of connection. This object uses the following integer values to denote the type of connection: <ul style="list-style-type: none"> <li>■ 1 other</li> <li>■ 2 P2P</li> </ul>
jnxAtmTrunkEncapsulation	jnxAtmTrunkEntry 3	Represents the ATM encapsulation type associated with the VC or trunk. This object uses the following integer values to denote the encapsulation type: <ul style="list-style-type: none"> <li>■ 1 other</li> <li>■ 2 atmCccCellRelay</li> </ul>
jnxAtmTrunkFlags	jnxAtmTrunkEntry 4	Represents the flags related to the trunk.
jnxAtmTrunkTotalDownTime	jnxAtmTrunkEntry 5	Indicates the total downtime (in seconds) for the trunk since the last reboot of the system.
jnxAtmTrunkInBytes	jnxAtmTrunkEntry 6	Indicates the number of bytes received on the trunk.
jnxAtmTrunkOutBytes	jnxAtmTrunkEntry 7	Indicates the number of bytes sent out on the trunk.
jnxAtmTrunkInPkts	jnxAtmTrunkEntry 8	Indicates the number of packets received on the trunk.
jnxAtmTrunkOutPkts	jnxAtmTrunkEntry 9	Indicates the number of packets sent out on the trunk.



**Table 145: jnxAtmTrunkTable** *(continued)*

Object	Object ID	Description
jnxAtmTrunkTailQueuePktDrops	jnxAtmTrunkEntry 10	Represents the number of packets that were dropped because of bandwidth constraints. This object indicates that the packets were queued to be transmitted at a rate faster than allowed.
jnxAtmTrunkInOAMF4AISCells	jnxAtmTrunkEntry 15	Indicates the number of OAM F4 cells that are received with AIS (Alarm Indication Signal) bit set.
jnxAtmTrunkOutOAMF4AISCells	jnxAtmTrunkEntry 16	Indicates the number of OAM F4 cells that are sent out with AIS bit set.



## Chapter 53

# Interpreting the Enterprise-Specific Configuration Management MIB

The enterprise-specific Configuration Management MIB, whose object identifier is {jnxMibs 18}, defines the objects that are used for managing the configuration of Juniper Networks products.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-cfgmgmt.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-cfgmgmt.txt).

This chapter contains the following sections:

- Text Conventions on page 533
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 534
- Rescue Configuration Change Management Objects on page 535
- Configuration Management Notifications on page 536

## Text Conventions

Table 146 on page 533 contains information on the text conventions used in the enterprise-specific configuration management MIB.

**Table 146: Text Conventions for Enterprise-Specific Configuration Management MIB**

Object	Description	Syntax
JnxCmCfgChgSource	Identifies the source of the configuration event.	<p>This object uses the following integer values:</p> <ul style="list-style-type: none"><li>■ 1—Other</li><li>■ 2—CLI</li><li>■ 3—JUNOScript</li><li>■ 4—Synchronize</li><li>■ 5—SNMP</li><li>■ 6—Button</li><li>■ 7—Autoinstall</li><li>■ 8—Unknown</li></ul>

**Table 146: Text Conventions for Enterprise-Specific Configuration Management MIB** *(continued)*

Object	Description	Syntax
JnxCmRescueCfgState	Represents the state of the rescue configuration.	This object uses the following integer values: <ul style="list-style-type: none"> <li>■ 1—Nonexistent</li> <li>■ 2—Updated</li> </ul>

## Configuration Change Management Objects and jnxCmCfgChgEventTable

The configuration change management objects listed in Table 147 on page 534 along with the objects in the `jnxCmCfgChgEventTable` provide you the configuration change history.

Each `jnxCmCfgChg`, whose object identifier is `{jnxCfgMgmt 1}`, contains the objects listed in Table 147 on page 534.

**Table 147: Configuration Change Management Objects**

Object	Object ID	Description
<code>jnxCmCfgChgLatestIndex</code>	<code>jnxCmCfgChg 1</code>	Contains the index used in <code>jnxCmCfgChgEventTable</code> to represent the latest configuration change event.
<code>jnxCmCfgChgLatestTime</code>	<code>jnxCmCfgChg 2</code>	Shows the value of <code>sysUpTime</code> at the time of the last configuration change. However, this object returns 0 if the management subsystem was reset after the last configuration change.
<code>jnxCmCfgChgLatestDate</code>	<code>jnxCmCfgChg 3</code>	Shows the date and time when the configuration was last modified.
<code>jnxCmCfgChgLatestSource</code>	<code>jnxCmCfgChg 4</code>	Shows the source from which the configuration event was triggered. This object uses <code>JnxCmCfChgSource</code> to represent the source of configuration event. For more information on <code>JnxCmCfChgSource</code> , see Table 146 on page 533.
<code>jnxCmCfgChgLatestUser</code>	<code>jnxCmCfgChg 5</code>	Shows the login name of the current user. This object returns a zero-length string if the user name is not available or not applicable.
<code>jnxCmCfgChgMaxEventEntries</code>	<code>jnxCmCfgChg 6</code>	Shows the maximum number of entries that <code>jnxCmCfgChgEventTable</code> can contain. Allowable range is 0 though 2147483647.  When the number of entries in <code>jnxCmCfgChgEventTable</code> exceeds the maximum value set for <code>jnxCmCfgChgMaxEventEntries</code> , the latest entry displaces the oldest entry in the table.

- `jnxCmCfgChgEventTable` on page 534

## *jnxCmCfgChgEventTable*

The `jnxCmCfgChgEventTable`, whose object identifier is `{jnxCmCfgChg 7}`, contains `jnxCmCfgChgEventEntry` that maps to the most recent configuration change events on

the router. The `jnxCmCfgChgMaxEventEntries` object discussed in the preceding section (Table 147 on page 534) controls the number of entries stored in `jnxCmCfgChgEventTable`.

Each `jnxCmCfgChgEventEntry`, whose object identifier is `{jnxCmCfgChgEventTable 1}`, contains the objects listed in Table 148 on page 535.

**Table 148: jnxCmCfgChgEventTable**

Object	Object ID	Description
<code>jnxCmCfgChgEventIndex</code>	<code>jnxCmCfgChgEventEntry 1</code>	Uniquely identifies a configuration change event. The SNMP process assigns monotonically increasing values to each event as it occurs. However, when the SNMP process is reset, the index values too are reset.
<code>jnxCmCfgChgEventTime</code>	<code>jnxCmCfgChgEventEntry 2</code>	Contains the value of <code>sysUpTime</code> when the event occurred.
<code>jnxCmCfgChgEventDate</code>	<code>jnxCmCfgChgEventEntry 3</code>	Contains the system date and time when the event occurred.
<code>jnxCmCfgChgEventSource</code>	<code>jnxCmCfgChgEventEntry 4</code>	Shows the source from which the configuration event was triggered. This object uses <code>JnxCmCfChgSource</code> to represent the source of configuration event. For more information on <code>JnxCmCfChgSource</code> , see Table 146 on page 533.
<code>jnxCmCfgChgEventUser</code>	<code>jnxCmCfgChgEventEntry 5</code>	Contains the name of the user who was logged in at the time of the event. Returns a zero-length string if the user name is not applicable or not available.
<code>jnxCmCfgChgEventLog</code>	<code>jnxCmCfgChgEventEntry 6</code>	Contains the log of the configuration event. Returns a zero-length string if no log is available.

## Rescue Configuration Change Management Objects

The `jnxCmRescueChg`, whose object identifier is `{jnxCfmgMgmt 2}`, contains information about changes to rescue configuration.

Table 149 on page 535 lists the objects associated with `jnxCmRescueChg`.

**Table 149: Rescue Configuration Change Management Objects**

Object	Object ID	Description
<code>jnxCmRescueChgTime</code>	<code>jnxCmRescueChg 1</code>	Contains the value of <code>sysUpTime</code> when the rescue configuration was last changed. If the management subsystem has been reset since the last configuration change, this object returns 0.
<code>jnxCmRescueChgDate</code>	<code>jnxCmRescueChg 2</code>	Contains the date and time when the rescue configuration was last changed.
<code>jnxCmRescueChgSource</code>	<code>jnxCmRescueChg 3</code>	Shows the source from which the rescue configuration event was triggered. This object uses <code>JnxCmCfChgSource</code> to represent the source of configuration event. For more information on <code>JnxCmCfChgSource</code> , see Table 146 on page 533.

**Table 149: Rescue Configuration Change Management Objects** *(continued)*

Object	Object ID	Description
jnxCmRescueChgUser	jnxCmRescueChg 4	Contains the name of the user who was logged in at the time of the event. Returns a zero-length string if the user name is not applicable or not available.
jnxCmRescueChgState	jnxCmRescueChg 5	Shows the current state of the rescue configuration. For more information on the different states of rescue configuration, see Table 146 on page 533.

## Configuration Management Notifications

The JUNOS software generates the following traps when a configuration or a rescue configuration event occurs:

- **jnxCmCfgChange**, whose object identifier is {jnxCmNotificationsPrefix 1}, contains jnxCmCfgChgEventTime, jnxCmCfgChgEventDate, jnxCmCfgChgEventSource, jnxCmCfgChgEventUser, and jnxCmCfgChgEventLog.



**NOTE:** Because configuration rollback is handled by the master management process that uses the **root** user ID, the **jnxCmCfgChgEventUser** object in the **jnxCmCfgChange** trap always returns **root** as the user name for configuration rollback events.

- **jnxCmRescueChange**, whose object identifier is {jnxCmNotificationsPrefix 2}, contains jnxCmRescueChgTime, jnxCmRescueChgDate, jnxCmRescueChgSource, jnxCmRescueChgUser, and jnxCmRescueChgState.

## Chapter 54

# Interpreting the Enterprise-Specific IPv4 MIB

The enterprise-specific IPv4 Management Information Base (MIB), whose object identifier is {**jnxMibs12**}, functions as an extension of the **ifTable** defined in *RFC 1573, IF MIB*, and defines the branches for IPV4 configuration.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipv4.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipv4.txt)

This chapter contains the following topic:

- **jnxIpv4AddrTable** on page 537

## jnxIpv4AddrTable

The **jnxIpv4AddrTable** defines the **jnxIpv4AddrEntry** and its attributes. Each **jnxIpv4AddrEntry** contains the objects listed in Table 150 on page 537.

**Table 150: jnxIpv4AddrTable**

Object	Object ID	Description
jnxIpv4AdEntIfIndex	jnxIpv4AddrEntry 1	A unique index value that identifies the interface with which a particular entry is associated. An interface identified by a particular value of <b>jnxIpv4AdEntIfIndex</b> is the same as the interface that is identified by the same value of <b>ifIndex</b> as defined in <i>RFC 1573</i> .
jnxIpv4AdEntAddr	jnxIpv4AddrEntry 2	The IP address of the interface with which the address information stored in this entry is associated.
jnxIpv4AdEntNetMask	jnxIpv4AddrEntry 3	The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0.
jnxIpv4AdEntBcastAddr	jnxIpv4AddrEntry 4	The least significant bit in the IP broadcast address used for sending datagrams on the logical interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcast addresses used by the entry on the logical interface.
jnxIpv4AdEntReasmMaxSize	jnxIpv4AddrEntry 5	The size of the largest IP datagram that this entry can reassemble from the incoming fragmented IP datagrams received on the interface.





## Chapter 55

# Interpreting the Enterprise-Specific Alarm MIB

The enterprise-specific Alarm MIB, whose object identifier is {jnxMibs 4}, contains information about alarms from the router chassis.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-alarm.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-chassis-alarm.txt).

This chapter contains the following sections:

- jnxAlarmRelayMode on page 539
- jnxYellowAlarms on page 539
- jnxRedAlarms on page 540

## jnxAlarmRelayMode

---

The jnxAlarmRelayMode, whose object identifier is {jnxCraftAlarms 1}, denotes the alarm relay mode of the craft interface panel for the yellow and red alarms. This object uses the following integer values:

- 1–Other: Other or unknown state
- 2–passOn: Alarms are passed on. The alarm relay is activated to pass on the yellow or red alarms to audible sirens or visual flashing devices.
- 3–cutOff: Alarms are turned off. Both the yellow and red alarms are cut off from the alarm relays and do not get passed on to audible sirens or visual flashing devices.



**NOTE:** Both the yellow and red alarms can be cut off from the alarm relay using a Alarm Cutoff/Lamp Test button on the front panel of the router chassis.

---

## jnxYellowAlarms

---

The jnxYellowAlarms, whose object identifier is {jnxCraftAlarms 2} contains the objects listed in Table 151 on page 540.

**Table 151: jnxYellowAlarms**

Object	Object ID	Description
jnxYellowAlarmState	jnxYellowAlarms 1	<p>Denotes the yellow alarm state on the craft interface panel of the router chassis. This object contains one of the following integer values:</p> <ul style="list-style-type: none"> <li>■ other-1: The alarm state is unknown.</li> <li>■ off-2: The yellow alarms are off. You can turn off the yellow alarms using the Alarm Cutoff/Lamp Test button on the craft interface panel of the router chassis.</li> <li>■ on-3: The yellow alarms are on.</li> </ul>
jnxYellowAlarmCount	jnxYellowAlarms 2	<p>Shows the number of currently active and non-silent yellow alarms.</p> <p><b>NOTE:</b> The value of this object is independent of the state of the Alarm Cutoff/Lamp Test button.</p>
jnxYellowAlarmLastChange	jnxYellowAlarms 3	<p>Shows the value of the <b>sysUp</b> time when the state of the yellow alarm last changed from on to off or vice versa. This object returns 0 if the alarm state has not changed since the <b>sysUp</b> time was reset last time, or if the value is unknown.</p>

## jnxRedAlarms

The **jnxRedAlarms**, whose object identifier is {**jnxCraftAlarms 3**}, contains the objects listed in Table 152 on page 540.

**Table 152: jnxRedAlarms**

Object	Object ID	Description
jnxRedAlarmState	jnxRedAlarms 1	<p>Denotes the state of red alarms on the craft interface panel of the router chassis. This object contains one of the following values:</p> <ul style="list-style-type: none"> <li>■ 1-other: The red alarm state is unknown.</li> <li>■ 2-off: The red alarm is turned off.</li> <li>■ 3-on: The red alarm is on. Typically, the red alarm is on when there is a system failure, power failure, or hardware malfunction, or when a threshold value is exceeded.</li> </ul>
jnxRedAlarmCount	jnxRedAlarms 2	<p>Shows the number of currently active and non-silent red alarms.</p> <p><b>NOTE:</b> The value of this object is independent of the state of the Alarm Cutoff/Lamp Test button.</p>
jnxRedAlarmLastChange	jnxRedAlarms 3	<p>Shows the value of the <b>sysUp</b> time when the red alarm last changed from on to off or vice versa. This object contains 0 value, if the alarm state has not changed since the <b>sysUp</b> time was reset last time, or if the value is unknown.</p>

## Chapter 56

# Interpreting the Enterprise-Specific Resource Reservation Protocol (RSVP) MIB

The enterprise-specific Resource Reservation Protocol (RSVP) MIB, whose object identifier is {jnxMibs 30}, contains information about RSVP-traffic engineering (TE) sessions that correspond to MPLS LSPs on transit routing platforms in the service provider core network.



**NOTE:** To collect information about MPLS statistics on transit routers, use the enterprise-specific RSVP MIB (`mib-jnx-rsvp.txt`) instead of the enterprise-specific MPLS MIB (`mib-jnx-mpls.txt`).

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rsvp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-rsvp.txt).

This chapter contains the following sections:

- `jnxRsvpSessionTable` on page 541

## **jnxRsvpSessionTable**

The `jnxRsvpSessionTable`, whose object identifier is {jnxRsvpOperation 1}, contains information about RSVP sessions. Each `jnxRsvpSessionEntry` (object identifier {jnxRsvpSessionTable 1}) is identified using a combination of two indexes, `jnxRsvpSessionName` and `jnxRsvpSessionIndex`. The `jnxRsvpSessionName` maps to the LSP name in MPLS entry, and can be used to correlate the `jnxRsvpSessionEntry` with `mplsLspEntry`. Because there can be multiple entries with the same RSVP session name, a secondary index, `jnxRsvpSessionIndex`, is used to uniquely identify each session in combination with the `jnxRsvpSessionName`.

Each `jnxRsvpSessionEntry` contains the objects listed in Table 153 on page 542.

**Table 153: jnxRsvpSessionTable**

Object	Object ID	Description
jnxRsvpSessionName	jnxRsvpSessionEntry 1	Contains the name of the RSVP session. This is the same as the LSP name in the <code>mplsLspEntry</code> and can contain up to 64 characters.
jnxRsvpSessionIndex	jnxRsvpSessionEntry 2	Uniquely identifies an RSVP session entry in combination with the <code>jnxRsvpSessionName</code> .
jnxRsvpSessionState	jnxRsvpSessionEntry 3	Shows the operational state of the RSVP session. This object contains one of the following integer values: <ul style="list-style-type: none"> <li>■ 1-Up</li> <li>■ 2-Down</li> </ul>
jnxRsvpSessionFrom	jnxRsvpSessionEntry 4	Contains the source IP address of the RSVP session.
jnxRsvpSessionTo	jnxRsvpSessionEntry 5	Contains the destination IP address of the RSVP session.
jnxRsvpSessionLspId	jnxRsvpSessionEntry 6	Contains the LSP ID of the sender for the RSVP session.
jnxRsvpSessionTunnelId	jnxRsvpSessionEntry 7	Contains the tunnel ID for the RSVP session.
jnxRsvpSessionPathType	jnxRsvpSessionEntry 8	Denotes the type of the path for the RSVP session. This object uses the following integer values to denote the path type: <ul style="list-style-type: none"> <li>■ 1-Primary</li> <li>■ 2-Secondary</li> <li>■ 3-unknown</li> </ul>
jnxRsvpSessionRole	jnxRsvpSessionEntry 9	Shows the role of an RSVP session with respect to the start and end points of the session. This object uses the following integer values to represent the role of the RSVP session: <ul style="list-style-type: none"> <li>■ 1-Ingress (source)</li> <li>■ 2-Transit (intermediate nodes)</li> <li>■ 3-Egress (destination)</li> </ul>
jnxRsvpSessionDiscontinuityTime	jnxRsvpSessionEntry 10	Shows the value of <code>sysUpTime</code> when either <code>jnxRsvpSessionMplsOctets</code> or <code>jnxRsvpSessionMplsPackets</code> counters experienced discontinuity. This object contains a zero value if no discontinuity occurred since the last initialization of the local management subsystem.
jnxRsvpSessionMplsOctets	jnxRsvpSessionEntry 11	Contains the number of MPLS octets that have been forwarded over the RSVP session. Because the MPLS statistics collection occurs at predefined intervals (default of 5 minutes), the value of this object may not reflect real-time statistics. This object is not updated if MPLS statistics collection is not enabled.
jnxRsvpSessionMplsPackets	jnxRsvpSessionEntry 12	Shows the number of MPLS packets that have been forwarded over the RSVP session. Because the MPLS statistics collection occurs at predefined intervals (default of 5 minutes), the value of this object may not reflect real-time statistics. This object is not updated if MPLS statistics collection is not enabled.

## Chapter 57

# Interpreting the Enterprise-Specific MPLS MIB

The enterprise-specific Multiprotocol Label Switching MIB, whose object identifier is {jnxMibs 2}, provides information about MPLS paths and defines MPLS notifications.

The table `mplsLspList` and the sequence for `mplsLspEntry` have been deprecated and replaced by a new table `mplsLspInfoList` to extend support for LSP names longer than 32 characters.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mpls.txt).

This chapter contains the following sections:

- MPLS Info Table on page 543
- MPLS Traffic Engineering (TE) Info Table on page 544
- `mplsAdminGroup` on page 544
- `mplsLspInfoList` on page 544
- Enterprise-Specific MPLS Traps on page 547

## MPLS Info Table

The `mplsInfo` table, whose object identifier is {mpls 1}, contains the objects listed in Table 154 on page 543.

**Table 154: mplsInfo**

Object	Object ID	Description
<code>mplsVersion</code>	<code>mplsInfo 1</code>	Shows the MPLS version number.
<code>mplsSignalingProto</code>	<code>mplsInfo 2</code>	Indicates the MPLS signaling protocol. This object uses the following integer values to denote the MPLS signaling protocol: <ul style="list-style-type: none"><li>■ 1–None</li><li>■ 2–Other</li><li>■ 3–RSVP</li><li>■ 4–LDP</li></ul>

**Table 154: mplsInfo** (continued)

Object	Object ID	Description
mplsConfiguredLsps	mplsInfo 3	Indicates the number of LSPs configured on the router.
mplsActiveLsps	mplsInfo 4	Indicates the number of active LSPs on the router.

## MPLS Traffic Engineering (TE) Info Table

The mplsTEInfo table, whose object identifier is {mpls 2}, contains the objects listed in Table 155 on page 544.

**Table 155: mplsTEInfo**

Object	Object ID	Description
mplsTEDistProtocol	mplsTEInfo 1	Indicates the Interior Gateway Protocol (IGP) used to distribute Traffic Engineering information and topology to each Label Switch Router (LSR) for automatic path computation. This object uses the following integer values to denote the protocols: <ul style="list-style-type: none"> <li>■ 1–None</li> <li>■ 2–IS-IS</li> <li>■ 3–OSPF</li> <li>■ 4–IS-IS – OSPF</li> </ul>
mplsAdminGroupList	mplsTEInfo 2	Contains the list of administrative groups configured on the router. Administrative groups are used to label links in the Traffic Engineering topology for specifying constraints (include and exclude) on LSP paths.

## mplsAdminGroup

The mplsAdminGroup, whose object identifier is {mplsAdminGroupList 1} contains mplsAdminGroupNumber and mplsAdminGroupName objects, and provides a mapping between the group number and name.

- mplsAdminGroupNumber–Whose object identifier is {mplsAdminGroup 1} contains an integer value from 0 through 31. This object indexes the mplsAdminGroup.
- mplsAdminGroupName–Whose object identifier is {mplsAdminGroup 2}, contains the name of the mplsAdminGroup. This object can contain a string of not more than 16 characters.

## mplsLsplInfoList

The mplsLsplInfoList, whose object identifier is {mpls 5} contains a list of Label Switched Paths (LSP) configured on the router. Each MplsLsplInfoEntry contains information about a particular LSP, and has the objects listed in Table 156 on page 545.

**Table 156: MplsLspInfoEntry**

Object	Object ID	Description
mplsLspInfoName	mplsLspInfoEntry 1	Contains the name of the LSP. This object can contain a string of not more than 64 characters.
mplsLspInfoState	mplsLspInfoEntry 2	Contains one of the following integer values to indicate the operational state of the LSP: <ul style="list-style-type: none"> <li>■ 1–Unknown</li> <li>■ 2–Up</li> <li>■ 3–Down</li> </ul>
mplsLspInfoOctets	mplsLspInfoEntry 3	Indicates the number of octets that have been forwarded over the current LSP active path. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object may not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsLspInfoPackets	mplsLspInfoEntry 4	Indicates the number of packets that have been forwarded over the current LSP active path. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object may not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsLspInfoAge	mplsLspInfoEntry 5	Indicates the time duration (in 10-millisecond intervals) since the inception of the LSP.
mplsLspInfoTimeUp	mplsLspInfoEntry 6	Indicates the total time (in 10-millisecond intervals) that the LSP has been operational. The percentage of up time can be calculated using the following formula: $\text{mplsLspInfoTimeUp} / \text{mplsLspInfoAge} \times 100$ .
mplsLspInfoPrimaryTimeUp	mplsLspInfoEntry 7	Indicates the total time (in 10-millisecond intervals) when the primary path of the LSP has been operational.
mplsLspInfoTransitions	mplsLspInfoEntry 8	Indicates the number of state transitions, from up to down and down to up, that the LSP has undergone.
mplsLspInfoLastTransition	mplsLspInfoEntry 9	Shows the time (in 10-millisecond intervals) since the last state transition occurred on the LSP.
mplsLspInfoPathChanges	mplsLspInfoEntry 10	Shows the number of path changes that occurred on the LSP. Every path change (path down, path up, and path change) generates a syslog entry or trap or both if the corresponding configuration is enabled.
mplsLspInfoLastPathChange	mplsLspInfoEntry 11	Indicates the time (in 10-millisecond intervals) since the last path change occurred on the LSP.
mplsLspInfoConfiguredPaths	mplsLspInfoEntry 12	Indicates the number of paths configured for the LSP.
mplsLspInfoStandbyPaths	mplsLspInfoEntry 13	Indicates the number of standby paths configured on the LSP.
mplsLspInfoOperationalPaths	mplsLspInfoEntry 14	Indicates the number of operational paths for the LSP. The value of this object includes the currently active path as well as the operational standby paths.
mplsLspInfoFrom	mplsLspInfoEntry 15	Contains the source IP address of the LSP.

**Table 156: MplsLsplInfoEntry** (continued)

Object	Object ID	Description
mplsLsplInfoTo	mplsLsplInfoEntry 16	Contains the destination IP address of the LSP.
mplsPathInfoName	mplsLsplInfoEntry 17	Shows the name of the active path for the LSP. If the path does not have a name, the <b>mplsLsplInfoEntry</b> objects listed in this table are invalid.
mplsPathInfoType	mplsLsplInfoEntry 18	<p>Contains one of the following integer values to denote the type of the active path:</p> <ul style="list-style-type: none"> <li>■ 1–Other</li> <li>■ 2–Primary</li> <li>■ 3–Standby</li> <li>■ 4–Secondary</li> </ul> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoExplicitRoute	mplsLsplInfoEntry 19	<p>Contains the explicit route used to set up the LSP. The explicit router can be one configured by the user or a generated route that satisfies the constraints set by the user.</p> <p>The value of this object is stored in the following format: xxx.xxx.xxx.xxx S/L, where S/L stands for Strict/Loose route. Each explicit route appears in a new line.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoRecordRoute	mplsLsplInfoEntry 20	<p>Shows the route actually used for the LSP as recorded by the signaling protocol.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoBandwidth	mplsLsplInfoEntry 21	<p>Indicates the configured bandwidth (in kbps) for the LSP.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoCOS	mplsLsplInfoEntry 22	<p>Indicates the class of service (CoS) configured for the path. If the value of this object is from 0 through 7, it goes in the 3-bit CoS field in the label. If the value is 255, the value in the CoS field of the label depends on other factors.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoInclude	mplsLsplInfoEntry 23	<p>Contains a configured set of colors represented by bit vector. For each link this path goes through, the link must have colors associated with the path, and the intersection of the link's colors and the <b>include</b> set must be set to a value other than null.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>
mplsPathInfoExclude	mplsLsplInfoEntry 24	<p>Contains a configured set of colors represented by bit vector. For each link the path goes through, the link must have colors associated with the path, and the intersection of the link's colors and the <b>exclude</b> set must be set to null.</p> <p><b>NOTE:</b> The value of this object is invalid if <b>mplsPathInfoName</b> is blank.</p>



**Table 156: MplsLspInfoEntry (continued)**

Object	Object ID	Description
mplsPathInfoSetupPriority	mplsLspInfoEntry 25	Indicates the set up priority configured for the path. This object contains integer values from 0 through 7.  <b>NOTE:</b> The value of this object is invalid if mplsPathInfoName is blank.
mplsPathInfoHoldPriority	mplsLspInfoEntry 26	Indicates the hold priority configured for the path. This object contains integer values from 0 through 7.  <b>NOTE:</b> The value of this object is invalid if mplsPathInfoName is blank.
mplsPathInfoProperties	mplsLspInfoEntry 27	Denotes the properties configured for the path. This value is represented as a bit map. The possible values are: <ul style="list-style-type: none"> <li>■ 1–Record-Route</li> <li>■ 2–Adaptive</li> <li>■ 4–CSPF</li> <li>■ 8–Mergeable</li> <li>■ 16–Preemptable</li> <li>■ 32–Preemptive</li> <li>■ 64–Fast-Reroute</li> </ul> <b>NOTE:</b> The value of this object is invalid if mplsPathInfoName is blank.

## Enterprise-Specific MPLS Traps

Table 157 on page 547 lists the enterprise-specific MPLS traps based on mplsLspInfoName.

**Table 157: MPLS Traps**

Object	Object ID	Description
mplsLspInfoUp	mplsLspTraps 1	Indicates that the LSP (mplsLspInfoName) is up. The current active path is represented by mplsPathInfoName.
mplsLspInfoDown	mplsLspTraps 2	Indicates that the LSP (mplsLspInfoName) is down because the current active path (mplsPathInfoName) has gone down.
mplsLspInfoChange	mplsLspTraps 3	Indicates that the LSP (mplsLspInfoName) has switched traffic to a new active path (mplsPathInfoName) without changing the state (up) before or after the switch.
mplsLspInfoPathDown	mplsLspTraps 4	Indicates that the specified path (mplsPathInfoName) for the LSP (mplsLspInfoName) has gone down.
mplsLspInfoPathUp	mplsLspTraps 5	Indicates that the specified path (mplsPathInfoName) for the LSP (mplsLspInfoName) has come up.



## Chapter 58

# Interpreting the Enterprise-Specific Host Resources MIB

The enterprise-specific Host Resources MIB, whose object identifier is {*jnxMibs* 31}, extends the *hrStorageTable* defined in RFC 2790, the standard Host Resources MIB, to include the *jnxHrStoragePercentUsed* object.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-hostresources.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-hostresources.txt)

This chapter contains the following section:

- *jnxHrStorageTable* on page 549

### ***jnxHrStorageTable***

---

The *jnxHrStorageTable*, whose object identifier is {*jnxHrStorage* 1}, contains *jnxHrStorageEntry*. Each *jnxHrStorageEntry* augments the *hrStorageEntry* to provide additional file system data, and contains the following object:

- *jnxHrStoragePercentUsed*—object identifier is {*jnxHrStorageEntry* 1}—Shows what percentage of the total storage space has been used.



Chapter 59

# **Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB**



## Chapter 60

# Interpreting the Enterprise-Specific MIMSTP MIB

The JUNOS software provides SNMP support for spanning-tree protocols on MX-series Ethernet Services routers.

The following standard and Juniper Networks enterprise-specific MIBs have been added to extend SNMP support to spanning-tree protocols:

- RFC 4188, *Definitions of Managed Objects for Bridges*—Supports 802.1d STP (1998) only.
- RFC 4318, *Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol*—Supports 802.1w and 802.1t extensions for RSTP.
- Juniper Networks Enterprise-Specific Multiple Instance Virtual Switch MIB (`mib-jnx-mimstp.txt`)—Supports 802.1s (2002) for MSTP.

The Juniper Networks enterprise-specific Multiple Instance Multiple Spanning Tree protocol (MIMSTP) management information base (MIB) (`mib-jnx-mimstp.txt`) provides information on multiple spanning-tree instances, that is routing instances of type Virtual Switch/Layer 2 control, also known as virtual contexts and associated VLANs.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mimstp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-mimstp.txt).

This chapter discusses the following topics:

- `jnxMIDot1sJuniperMstTable` on page 553
- Juniper Networks MSTI Bridge Table on page 558
- `jnxMIMstVlanInstanceMappingTable` on page 560
- `jnxMIMstCistPortTable` on page 561
- `jnxMIMstMstiPortTable` on page 564
- Juniper Networks Enterprise-Specific MIMSTP Traps on page 566

## **`jnxMIDot1sJuniperMstTable`**

---

The `jnxMIDot1sJuniperMstTable` provides MSTP module parameters for a given virtual context. Table 158 on page 554 lists the objects in the `jnxMIDot1sJuniperMstTable`.

**Table 158: jnxMIDot1sJuniperMstTable**

Object	Object Identifier	Description
jnxMIDot1sJuniperMstContextId	jnxMIDot1sJuniperMstEntry 1	Identifies the virtual context.
jnxMIMstSystemControl	jnxMIDot1sJuniperMstEntry 2	Indicates the status of MST on the ports of the device. The value <b>start (1)</b> indicates that MIMST is active on all ports of the device; the value <b>shutdown (2)</b> indicates that MIMST is shut down on all ports of the device.
jnxMIMstModuleStatus	jnxMIDot1sJuniperMstEntry 3	Indicates whether MST is enabled or disabled on the ports. When set to value 1, <b>enabled</b> , MST is enabled on all ports of the device; when set to value 2, <b>disabled</b> , MIMST is disabled on all ports.  <b>NOTE:</b> The object can be set to <b>enabled(1)</b> only when <b>jnxMIMstSystemControl</b> is set to <b>start</b> .
jnxMIMstMaxMstInstanceNumber	jnxMIDot1sJuniperMstEntry 4	Indicates the maximum number of spanning-tree instances allowed on the bridge.
jnxMIMstNoOfMstiSupported	jnxMIDot1sJuniperMstEntry 5	Indicates the maximum number of spanning-tree instances that can be supported on the bridge.
jnxMIMstMaxHopCount	jnxMIDot1sJuniperMstEntry 6	Returns the Maximum Hop Count value.
jnxMIMstBrgAddress	jnxMIDot1sJuniperMstEntry 7	Indicates the MAC address used by the bridge, which forms a unique bridge identifier when combined with <b>jnxMIMstCistBridgePriority</b> or <b>jnxMIMstMstiBridgePriority</b> .
jnxMIMstCistRoot	jnxMIDot1sJuniperMstEntry 8	Indicates the bridge identifier of the root of the common spanning tree as determined by the Spanning Tree Protocol. This value is used as the CIST Root Identifier parameter in all configuration bridge PDUs originating at this node.
jnxMIMstCistRegionalRoot	jnxMIDot1sJuniperMstEntry 9	Indicates the bridge identifier of the root of the multiple spanning-tree region as determined by the Spanning Tree Protocol. This value is used as the CIST Regional Root Identifier parameter in all configuration bridge PDUs originating at this node.
jnxMIMstCistRootCost	jnxMIDot1sJuniperMstEntry 10	Indicates the cost of the path to the CIST root from this bridge.



**Table 158: jnxMIDot1sJuniperMstTable** (continued)

Object	Object Identifier	Description
jnxMIMstCistRegionalRootCost	jnxMIDot1sJuniperMstEntry 11	Indicates the cost of the path to the CIST regional root from this bridge.
jnxMIMstCistRootPort	jnxMIDot1sJuniperMstEntry 12	Indicates the port number of the port that offers the lowest path cost from this bridge to the CIST Root Bridge.
jnxMIMstCistBridgePriority	jnxMIDot1sJuniperMstEntry 13	Indicates the value of the writable portion of the bridge identifier. The values set for Bridge Priority must be in multiples of 4096.
jnxMIMstCistBridgeMaxAge	jnxMIDot1sJuniperMstEntry 14s	Indicates the value that a bridge uses for MaxAge when the bridge is acting as the root.
jnxMIMstCistBridgeForwardDelay	jnxMIDot1sJuniperMstEntry 15	Indicates the value that the bridge uses for ForwardDelay when this bridge is acting as the root. Note that 802.1D specifies that the range for this parameter is related to the value of <b>BridgeMaxAge</b> .
jnxMIMstCistHoldTime	jnxMIDot1sJuniperMstEntry 16	Sets the interval between transmitting two configuration bridge PDUs.
jnxMIMstCistMaxAge	jnxMIDot1sJuniperMstEntry 17	Sets the maximum age of Spanning Tree Protocol (STP) information learned on the ports. The STP information is discarded when the age exceeds the set limit.
jnxMIMstCistForwardDelay	jnxMIDot1sJuniperMstEntry 18	Indicates the time period during which a port stays in a particular state before moving to the next state; for example, from spanning to forwarding state.
jnxMIMstMstpUpCount	jnxMIDot1sJuniperMstEntry 19	Shows the number of times the MSTP Module has been enabled on the bridge.
jnxMIMstMstpDownCount	jnxMIDot1sJuniperMstEntry 20	Shows the number of times the MSTP Module has been disabled on the bridge.
jnxMIMstPathCostDefaultType	jnxMIDot1sJuniperMstEntry 21	Shows the version of the spanning tree default path costs that are to be used by the bridge. <ul style="list-style-type: none"> <li>■ A value of 8021d1998(1) uses the 16-bit default path costs from IEEE Std. 802.1D-1998.</li> <li>■ A value of stp8021t2001(2) uses the 32-bit default path costs from IEEE Std. 802.1t.</li> </ul>

**Table 158: jnxMIDot1sJuniperMstTable** (continued)

Object	Object Identifier	Description
jnxMIMstDebug	jnxMIDot1sJuniperMstEntry 23	<p>Enables debug statements in the MSTP module. A four-byte integer can be used to set the level of debugging.</p> <p>The bit position maps to the following levels of debugging:</p> <ul style="list-style-type: none"> <li>■ 0: Init and shutdown debug statements</li> <li>■ 1: Management debug statements</li> <li>■ 2: Memory-related debug statements</li> <li>■ 3: BPDU-related debug statements</li> <li>■ 4: Event handling debug statements</li> <li>■ 5: Time module debug statements</li> <li>■ 6: Port information SEM debug statements</li> <li>■ 7: Port receive SEM debug statements (valid in the case of MSTP alone)</li> <li>■ 8: Role selection SEM debug statements</li> <li>■ 9: Role transition SEM debug statements</li> <li>■ 10: State transition SEM debug statements</li> <li>■ 11: Protocol migration SEM debug statements</li> <li>■ 12: Topology change SEM debug statements</li> <li>■ 13: Port transmit SEM debug statements</li> <li>■ 14: Bridge detection SEM debug statements</li> <li>■ 15: All failure debug statements</li> <li>■ 16: Redundancy code flow debug statements</li> </ul> <p>The rest of the bits remain unused. You can use a combination of debug levels to generate debug statements of multiple debug levels.</p> <p><b>NOTE:</b> Debug options and trace options are mutually exclusive. When the debug option is set, the trace option is set to 0.</p>

**Table 158: jnxMIDot1sJuniperMstTable** (continued)

Object	Object Identifier	Description
jnxMIMstForceProtocolVersion	jnxMIDot1sJuniperMstEntry 24	<p>Indicates the version of the Spanning Tree Protocol that is running on the bridge.</p> <ul style="list-style-type: none"> <li>■ <b>stpCompatible(0)</b>: Represents Spanning Tree Protocol specified in IEEE 802.1D.</li> <li>■ <b>rstp(2)</b>: Represents the Rapid Spanning Tree protocol specified in IEEE 802.1w.</li> <li>■ <b>mstp(3)</b>: Represents the Multiple Spanning Tree protocol specified in IEEE 802.1s.</li> </ul>
jnxMIMstTxHoldCount	jnxMIDot1sJuniperMstEntry 25	Indicates the value that the port transmit state machine uses to limit the maximum transmission rate.
jnxMIMstMstiConfigIdSel	jnxMIDot1sJuniperMstEntry 26	Indicates the Configuration Identifier Format Selector that the bridge uses.
jnxMIMstMstiRegionName	jnxMIDot1sJuniperMstEntry 27	Indicates the name of the region's configuration. By default, the region name and the MAC address of the bridge are the same.
jnxMIMstMstiRegionVersion	jnxMIDot1sJuniperMstEntry 28	Indicates the version of the multiple-spanning tree region.
jnxMIMstMstiConfigDigest	jnxMIDot1sJuniperMstEntry 29	Indicates the configuration digest value for the multiple-spanning tree region.
jnxMIMstBufferOverFlowCount	jnxMIDot1sJuniperMstEntry 30	Indicates the number of times buffer overflows or failures have occurred. This event generates a trap.
jnxMIMstMemAllocFailureCount	jnxMIDot1sJuniperMstEntry 31	Indicates the number of times memory allocation failures have occurred. This event generates a trap.
jnxMIMstRegionConfigChangeCount	jnxMIDot1sJuniperMstEntry 32	Indicates the number of times a Region Configuration Identifier Change was detected. This event generates a trap.
jnxMIMstCistBridgeRoleSelectionSemState	jnxMIDot1sJuniperMstEntry 33	Indicates the current state of the Port Role Selection State Machine of the bridge in a common spanning tree context
jnxMIMstCistTimeSinceTopologyChange	jnxMIDot1sJuniperMstEntry 34	Indicates the time since the TcWhile Timer for any port of the Bridge was non-zero for the common spanning-tree context.

**Table 158: jnxMIDot1sJuniperMstTable** (continued)

Object	Object Identifier	Description
jnxMIMstCistTopChanges	jnxMIDot1sJuniperMstEntry 35	Indicates the number of times when there was at least one non-zero TcWhile Timer on the bridge for the common spanning-tree context.
jnxMIMstCistNewRootBridgeCount	jnxMIDot1sJuniperMstEntry 36	Indicates how many times the bridge has detected a root bridge change for a common-spanning tree context. This event generates a trap.
jnxMIMstCistHelloTime	jnxMIDot1sJuniperMstEntry 37	Specifies the interval between the transmission of configuration BPDUs by this node on any port when it is either the root of the spanning tree or trying to become the root.
jnxMIMstCistBridgeHelloTime	jnxMIDot1sJuniperMstEntry 38	Specifies the interval between the transmission of configuration bridge PDUs by this node.
jnxMIMstCistDynamicPathcostCalculation	jnxMIDot1sJuniperMstEntry 39	Indicates whether the dynamic path cost calculation is allowed. If set to true, path cost is calculated dynamically from the port speed; otherwise the link speed at the time of port creation is used for calculating the path cost. In both cases, the user has configured a path cost for the port that will be used. By default, dynamic path cost calculation is set to false.

## Juniper Networks MSTI Bridge Table

The jnxMIMstMstiBridgeTable provides information on a bridge that belongs to a given spanning-tree instance (MSTI). Each jnxMIMstMstiBridgeEntry points to a bridge in the spanning-tree instance and has the objects listed in Table 159 on page 558.

**Table 159: jnxMIMstMstiBridgeTable**

Object	Object Identifier	Description
jnxMIMstMstiInstanceIndex	jnxMIMstMstiBridgeEntry 1	Identifies the spanning-tree instance to which the information belongs.
jnxMIMstMstiBridgeRegionalRoot	jnxMIMstMstiBridgeEntry 2	Indicates the MSTI Regional Root Identifier value for the Instance. This value is used as the Regional Root Identifier parameter in all the configuration bridge PDUs originated by this node.

**Table 159: jnxMIMstMstiBridgeTable** (continued)

Object	Object Identifier	Description
jnxMIMstMstiBridgePriority	jnxMIMstMstiBridgeEntry 3	Indicates the writable portion of the MSTI Bridge Identifier that comprises the first two octets. The values that are set for Bridge Priority must be in multiples of 4096.
jnxMIMstMstiRootCost	jnxMIMstMstiBridgeEntry 4	Indicates the cost of the path to the MSTI Regional Root as calculated by the bridge.
jnxMIMstMstiRootPort	jnxMIMstMstiBridgeEntry 5	Indicates the port number of the port that offers the lowest path cost from the bridge to the MSTI Region Root Bridge.
jnxMIMstMstiTimeSinceTopologyChange	jnxMIMstMstiBridgeEntry 6	Indicates the time (in hundredths of a second) since the TcWhile Timer for any port on this bridge was non-zero for this spanning-tree instance.
jnxMIMstMstiTopChanges	jnxMIMstMstiBridgeEntry 7	Indicates the number of times when there was at least one non-zero TcWhile Timer on the bridge for the spanning-tree instance.
jnxMIMstMstiNewRootBridgeCount	jnxMIMstMstiBridgeEntry 8	Indicates the number of times the bridge has detected a root bridge change for the spanning-tree instance. This event generates a trap.
jnxMIMstMstiBridgeRoleSelectionSemState	jnxMIMstMstiBridgeEntry 9	Shows the current state of the Port Role Selection State Machine for the spanning-tree instance of this bridge.
jnxMIMstInstanceUpCount	jnxMIMstMstiBridgeEntry 10	Indicates the number of times a new spanning-tree instance has been created. This counter is incremented whenever a new spanning-tree instance is created and also whenever a VLAN is mapped to the instance. This event generates a trap.
jnxMIMstInstanceDownCount	jnxMIMstMstiBridgeEntry 11	Indicates the number of times a spanning-tree instance has been deleted. This counter is incremented whenever a spanning tree instance is deleted and also whenever a VLAN is unmapped from the instance. This event generates a trap.
jnxMIMstOldDesignatedRoot	jnxMIMstMstiBridgeEntry 12	Indicates the bridge identifier of the old root of the spanning-tree instance as determined by the Spanning Tree Protocol.

## jnxMIMstVlanInstanceMappingTable

The `jnxMIMstVlanInstanceMappingTable` contains information on the mapping between each instance of MSTP and associated VLANs. Each `jnxMIMstVlanInstanceMappingEntry` indicates the status and properties of a specific MSTP instance-VLAN mapping and has the objects listed in Table 160 on page 560.

**Table 160: jnxMIMstVlanInstanceMappingTable**

Object	Object Identifier	Description
<code>jnxMIMstInstanceIndex</code>	<code>jnxMIMstVlanInstanceMappingEntry 1</code>	Identifies a multiple spanning-tree instance using an arbitrary integer from 1 through the value of <code>Max Instance Number</code> .
<code>jnxMIMstMapVlanIndex</code>	<code>jnxMIMstVlanInstanceMappingEntry 2</code>	Indicates that the VLAN ID is mapped to the multiple spanning-tree instance specified.
<code>jnxMIMstUnMapVlanIndex</code>	<code>jnxMIMstVlanInstanceMappingEntry 3</code>	Indicates that the VLAN ID is unmapped from the spanning-tree instance to which it was mapped.
<code>jnxMIMstInstanceVlanMapped</code>	<code>jnxMIMstVlanInstanceMappingEntry 6</code>	Represents a string of octets that contain one bit per VLAN. The first octet corresponds to VLANs with <code>VlanIndex</code> values 1 through 8; the second octet to VLANs 9 through 16, and so on. The most significant bit of each octet corresponds to the lowest <code>VlanIndex</code> value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.
<code>jnxMIMstInstanceVlanMapped2k</code>	<code>jnxMIMstVlanInstanceMappingEntry 7</code>	Represents a string of octets that contain one bit per VLAN for VLANs with <code>VlanIndex</code> values from 1024 through 2047. The first octet corresponds to VLANs with <code>VlanIndex</code> values 1024 through 1031; the second octet to VLANs 1032 through 1039, and so on. The most significant bit of each octet corresponds to the lowest <code>VlanIndex</code> value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.

**Table 160: jnxMIMstVlanInstanceMappingTable** (continued)

Object	Object Identifier	Description
jnxMIMstInstanceVlanMapped3k	jnxMIMstVlanInstanceMappingEntry 8	Represents a string of octets that contain one bit per VLAN for VLANs with <b>VlanIndex</b> values from 2048 through 3071. The first octet corresponds to VLANs with <b>VlanIndex</b> values 2048 through 2055; the second octet to VLANs 2056 through 2063, and so on. The most significant bit of each octet corresponds to the lowest <b>VlanIndex</b> value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.
jnxMIMstInstanceVlanMapped4k	jnxMIMstVlanInstanceMappingEntry 9	Represents a string of octets that contain one bit per VLAN for VLANs with <b>VlanIndex</b> values from 3072 through 4095. The first octet corresponds to VLANs with <b>VlanIndex</b> values 3072 through 3079; the second octet to VLANs 3080 through 3087, and so on. The most significant bit of each octet corresponds to the lowest <b>VlanIndex</b> value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.

## jnxMIMstCistPortTable

The **jnxMIMstCistPortTable** contains the information maintained by the ports of Common and Internal Spanning Tree Protocol. Table 161 on page 561 lists the parameters maintained by each **jnxMIMstCistPortEntry**.

**Table 161: jnxMIMstCistPortTable**

Object	Object Identifier	Description
jnxMIMstCistPort	jnxMIMstCistPortEntry 1	Specifies the port number of the port to which this entry is mapped.
nxMIMstCistPortPathCost	jnxMIMstCistPortEntry 2	Indicates the contribution of this port to the path cost of paths towards the CIST root that includes this port.
jnxMIMstCistPortPriority	jnxMIMstCistPortEntry 3	Contains the four most significant bits of the Port Identifier of the spanning-tree instance that can be modified by setting the <b>CistPortPriority</b> value. The values that are set for Port Priority must be in multiples of 16.
jnxMIMstCistPortDesignatedRoot	jnxMIMstCistPortEntry 4	Specifies the unique Bridge Identifier that is recorded as the CIST root in the configuration BPDUs.

**Table 161: jnxMIMstCistPortTable** (continued)

Object	Object Identifier	Description
jnxMIMstCistPortDesignatedBridge	jnxMIMstCistPortEntry 5	Specifies the unique Bridge Identifier of the bridge that is considered as the designated bridge for the port's segment.
jnxMIMstCistPortDesignatedPort	jnxMIMstCistPortEntry 6	Indicates the port identifier of the port on the designated bridge for this port's segment.
jnxMIMstCistPortAdminP2P	jnxMIMstCistPortEntry 7	Indicates the administrative point-to-point status of the LAN segment attached to this port. <ul style="list-style-type: none"> <li>■ A value of <b>forceTrue(0)</b> indicates that this port must be treated as if it were connected to a point-to-point link.</li> <li>■ A value of <b>forceFalse(1)</b> indicates that this port should be treated as having a shared media connection.</li> <li>■ A value of <b>auto(2)</b> indicates that this port is considered to have a point-to-point link if it is an aggregator and all of its members are aggregatable, or if the MAC entity is configured for full duplex operation, either through autonegotiation or by management.</li> </ul>
jnxMIMstCistPortOperP2P	jnxMIMstCistPortEntry 8	Indicates the operational point-to-point status of the LAN segment that is attached to this port. It indicates whether a port is considered to have a point-to-point connection or not. The value is determined by management or by autodetection, as described in the <b>jnxMIMstCistPortAdminP2P</b> object.
jnxMIMstCistPortAdminEdgeStatus	jnxMIMstCistPortEntry 9	Specifies the administrative value of the <b>EdgePort</b> parameter. A value of <b>TRUE(1)</b> indicates that this port must be treated as an edge port, and a value of <b>FALSE(2)</b> indicates that this port should be treated as a non-edge port.
jnxMIMstCistPortOperEdgeStatus	jnxMIMstCistPortEntry 10	Specifies the operational value of the <b>EdgePort</b> parameter. The object is initialized to the value of <b>jnxMIMstCistPortAdminEdgeStatus</b> and is set <b>FALSE</b> on reception of a BPDU.
jnxMIMstCistPortState	jnxMIMstCistPortEntry 12	Shows the current state of the port as defined by the Common Spanning Tree Protocol.
jnxMIMstCistPortForwardTransitions	jnxMIMstCistPortEntry 14	Indicates the number of times this port has transitioned to the forwarding state.
jnxMIMstCistPortRxMstBpduCount	jnxMIMstCistPortEntry 15	Indicates the number of multiple spanning-tree BPDUs that are received on this port.
jnxMIMstCistPortRxRstBpduCount	jnxMIMstCistPortEntry 16	Indicates the number of rapid spanning-tree BPDUs that are received on this port.
jnxMIMstCistPortRxConfigBpduCount	jnxMIMstCistPortEntry 17	Indicates the number of configuration BPDUs that are received on the port.



**Table 161: jnxMIMstCistPortTable** (continued)

Object	Object Identifier	Description
jnxMIMstCistPortRxTcnBpduCount	jnxMIMstCistPortEntry 18	Indicates the number of topology change notification BPDUs that are received on the port.
jnxMIMstCistPortTxMstBpduCount	jnxMIMstCistPortEntry 19	Indicates the number of multiple spanning-tree BPDUs that are transmitted from the port.
jnxMIMstCistPortTxRstBpduCount	jnxMIMstCistPortEntry 20	Indicates the number of rapid spanning-tree BPDUs that are transmitted from the port.
jnxMIMstCistPortTxConfigBpduCount	jnxMIMstCistPortEntry 21	Indicates the number of configuration BPDUs that are transmitted from the port.
jnxMIMstCistPortTxTcnBpduCount	jnxMIMstCistPortEntry 22	Indicates the number of TCN BPDUs that are transmitted from the port.
jnxMIMstCistPortInvalidMstBpduRxCount	jnxMIMstCistPortEntry 23	Indicates the number of invalid MST BPDUs that are received on the port.
jnxMIMstCistPortInvalidRstBpduRxCount	jnxMIMstCistPortEntry 24	Indicates the number of invalid RST BPDUs that are received on the port.
jnxMIMstCistPortInvalidConfigBpduRxCount	jnxMIMstCistPortEntry 25	Indicates the number of invalid configuration BPDUs that are received on the port.
jnxMIMstCistPortInvalidTcnBpduRxCount	jnxMIMstCistPortEntry 26	Indicates the number of invalid TCN BPDUs that are received on the port.
jnxMIMstCistPortTransmitSemState	jnxMIMstCistPortEntry 27	Indicates the current state of the port transmit state machine.
jnxMIMstCistPortReceiveSemState	jnxMIMstCistPortEntry 28	Indicates the current state of the port receive state machine.
jnxMIMstCistPortProtMigrationSemState	jnxMIMstCistPortEntry 29	Indicates the current state of the port protocol migration state machine.
jnxMIMstCistProtocolMigrationCount	jnxMIMstCistPortEntry 30	Indicates the number of times the port has migrated from one Spanning Tree Protocol version to another. The relevant protocols are STP-COMPATIBLE and RSTP/MSTP. This event generates a trap.
jnxMIMstCistPortDesignatedCost	jnxMIMstCistPortEntry 31	Indicates the path cost of the designated port of the segment connected to this port.
jnxMIMstCistPortRegionalRoot	jnxMIMstCistPortEntry 32	Specifies the unique Bridge Identifier of the bridge recorded as the CIST Regional Root Identifier in the configuration BPDUs transmitted.
jnxMIMstCistPortRegionalPathCost	jnxMIMstCistPortEntry 33	Specifies the contribution of this port to the path cost of paths towards the CIST regional root that includes this port.
jnxMIMstCistSelectedPortRole	jnxMIMstCistPortEntry 34	Indicates the selected port role of the port for this spanning-tree instance.

**Table 161: jnxMIMstCistPortTable** (continued)

Object	Object Identifier	Description
jnxMIMstCistCurrentPortRole	jnxMIMstCistPortEntry 35	Specifies the current port role of the port for this spanning-tree instance.
jnxMIMstCistPortInfoSemState	jnxMIMstCistPortEntry 36	Indicates the current state of the port information state machine for this port in this spanning-tree context.
jnxMIMstCistPortRoleTransitionSemState	jnxMIMstCistPortEntry 37	Indicates the current state of the port role transition state machine for this port in this spanning tree context.
jnxMIMstCistPortStateTransitionSemState	jnxMIMstCistPortEntry 38	Indicates the current state of the port state transition state machine for this port in this spanning-tree context.
jnxMIMstCistPortTopologyChangeSemState	jnxMIMstCistPortEntry 39	Indicates the current state of the topology change state machine for this port in this spanning-tree context.
jnxMIMstCistPortHelloTime	jnxMIMstCistPortEntry 40	Indicates the interval between the transmission of configuration bridge PDUs on this port.
jnxMIMstCistPortOperVersion	jnxMIMstCistPortEntry 41	Indicates one of the following operational modes of the port: MSTP, RSTP, and STP-compatible.
jnxMIMstCistPortEffectivePortState	jnxMIMstCistPortEntry 42	Indicates the effective operational state of the port for CIST. This values is set to TRUE only when the port is operationally up in the Interface level and Protocol level for CIST. This value is set to FALSE for all other instances.
jnxMIMstCistPortAutoEdgeStatus	jnxMIMstCistPortEntry 43	Indicates one of the following states: <ul style="list-style-type: none"> <li>■ TRUE(1) when the detection of a port as edge post happens automatically</li> <li>■ FALSE(2) when this feature is disabled.</li> </ul>

## jnxMIMstMstiPortTable

The jnxMIMstMstiPortTable contains information maintained by the non-CIST ports for each spanning tree instance. Each jnxMIMstMstiPortEntry contains the parameters listed in Table 162 on page 564.

**Table 162: jnxMIMstMstiPortTable**

Object	Object Identifier	Description
jnxMIMstMstiPort	jnxMIMstMstiPortEntry 1	Specifies the port number of the port to which this entry maps.
jnxMIMstMstiPortPathCost	jnxMIMstMstiPortEntry 2	Shows the contribution of this port to the path cost of paths towards the MSTI root that includes this port.

**Table 162: jnxMIMstMstiPortTable** (continued)

Object	Object Identifier	Description
jnxMIMstMstiPortPriority	jnxMIMstMstiPortEntry 3	Enables you to specify the four most significant bits of the Port Identifier for a given spanning-tree instance that can be modified independently for each spanning-tree instance supported by the bridge. The values that are set for Port Priority must be in multiples of 16.
jnxMIMstMstiPortDesignatedRoot	jnxMIMstMstiPortEntry 4	Indicates the unique Bridge Identifier of the bridge recorded as the MSTI regional root in the configuration BPDUs transmitted.
jnxMIMstMstiPortDesignatedBridge	jnxMIMstMstiPortEntry 5	Indicates the unique Bridge Identifier of the bridge which this port considers to be the designated bridge for the port's segment.
jnxMIMstMstiPortDesignatedPort	jnxMIMstMstiPortEntry 6	Indicates the port identifier of the port on the designated bridge for this port's segment.
jnxMIMstMstiPortState	jnxMIMstMstiPortEntry 7	Indicates the current state of the port as defined by the Multiple Spanning Tree protocol. A port which is in forwarding state in one instance can be in discarding (blocking) state in another instance.
jnxMIMstMstiPortForwardTransitions	jnxMIMstMstiPortEntry 9	Indicates the number of times this port has transitioned to the forwarding state for specific instance.
jnxMIMstMstiPortReceivedBPDUs	jnxMIMstMstiPortEntry 10	Indicates the number of BPDUs received by this port for this spanning-tree instance.
jnxMIMstMstiPortTransmittedBPDUs	jnxMIMstMstiPortEntry 11	Indicates the number of BPDUs transmitted on this port for this spanning tree instance.
jnxMIMstMstiPortInvalidBPDUsRcvd	jnxMIMstMstiPortEntry 12	Indicates the number of invalid BPDUs received on this port for this spanning-tree instance.
jnxMIMstMstiPortDesignatedCost	jnxMIMstMstiPortEntry 13	Indicates the path cost of the designated port of the segment connected to this port.
jnxMIMstMstiSelectedPortRole	jnxMIMstMstiPortEntry 14	Indicates the selected Port Role of the port for this spanning-tree instance.
jnxMIMstMstiCurrentPortRole	jnxMIMstMstiPortEntry 15	Indicates the current Port Role of the port for this spanning-tree instance.
jnxMIMstMstiPortInfoSemState	jnxMIMstMstiPortEntry 16	Shows the current state of the port information state machine for this port in this spanning-tree context.
jnxMIMstMstiPortRoleTransitionSemState	jnxMIMstMstiPortEntry 17	Shows the current state of the port role transition state machine for this port in this spanning-tree context.
jnxMIMstMstiPortStateTransitionSemState	jnxMIMstMstiPortEntry 18	Shows the current state of the port state transition state machine for this port in this spanning-tree context.
jnxMIMstMstiPortTopologyChangeSemState	jnxMIMstMstiPortEntry 19	Shows the current state of the topology change state machine for this port in this spanning tree context.

**Table 162: jnxMIMstMstiPortTable** (continued)

Object	Object Identifier	Description
jnxMIMstMstiPortEffectivePortState	jnxMIMstMstiPortEntry 20	Shows the effective operational state of the port for the specific instance. The value is set to TRUE only when the port is operationally up in the interface level and protocol level for the specific instance. This is set to be FALSE at all other times.

## Juniper Networks Enterprise-Specific MIMSTP Traps

Table 163 on page 566 lists the Juniper Networks enterprise-specific MIMSTP traps.

**Table 163: Juniper Networks Enterprise-Specific MIMSTP Traps**

Object	Object Identifier	Description
jnxMIMstGenTrap	jnxMIMstTraps 1	Generated when any of the general events such as protocol up or protocol down occurs.
jnxMIMstErrTrap	jnxMIMstTraps 2	Generated when any of the error events such as memory failure, buffer failure, protocol migration, or new root or topology change occurs.
jnxMIMstNewRootTrap	jnxMIMstTraps 3	Generated when a new root bridge is selected in the topology. The jnxMIMstNewRootTrap indicates that the sending agent has become the new root of the spanning tree; the trap is sent by a bridge soon after its election as the new root.
jnxMIMstTopologyChgTrap	jnxMIMstTraps 4	Generated when a topology change is detected.
jnxMIMstProtocolMigrationTrap	jnxMIMstTraps 5	Generated when a port protocol migration happens on the port.
jnxMIMstInvalidBpduRxdTrap	jnxMIMstTraps 6	Generated when an invalid packet is received for bpdus/stp/rstp/maximum age/forward delay/hello time.
jnxMIMstRegionConfigChangeTrap	jnxMIMstTraps 7	Generated when the multiple spanning-tree region's configuration identifier changes.

## Chapter 61

# Interpreting the Enterprise-Specific L2ALD MIB

The enterprise-specific Layer 2 Address Learning Daemon (L2ALD) Management Information Base (MIB), whose object identifier is {jnxl2aldMibRoot 1}, contains information about Layer 2 addresses and defines L2ALD traps.

The L2ALD MIB has the following two branches:

- jnxl2aldNotification, whose object identifier is {jnxl2aldMib 0}
- jnxl2aldObjects, whose object identifier is {jnxl2aldMib 1}

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnxl2ald.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnxl2ald.txt).

This chapter contains the following sections:

- jnxl2aldInterfaceTable on page 567
- MAC Address Limit Traps on page 568

## jnxl2aldInterfaceTable

The jnxl2aldInterfaceTable, whose object identifier is {jnxl2aldObjects 1}, contains objects that control the MAC address limit on each interface. Each jnxl2aldEntry (object identifier: {jnxl2aldInterfaceTable 1}) in the jnxl2aldInterfaceTable contains the objects listed in Table 164 on page 567.

**Table 164: jnxl2aldInterfaceTable**

Object	Object ID	Description
jnxl2aldIntfLogicalRouter	jnxl2aldEntry 1	Identifies the logical system with which the jnxl2aldEntry is associated.
jnxl2aldIntfRoutingInst	jnxl2aldEntry 2	Identifies the routing instance with which the jnxl2aldEntry is associated.
jnxl2aldIntfBridgeDomain	jnxl2aldEntry 3	Identifies the bridge domain with which the jnxl2aldEntry is associated.

**Table 164: jnxl2aldInterfaceTable** (continued)

Object	Object ID	Description
jnxl2aldIntfMacLimit	jnxl2aldEntry 4	Defines the MAC address limit for physical interface associated with the jnxl2aldEntry.
<b>Scalar Objects for Notifications</b>		
jnxl2aldRoutingInst	jnxl2aldObjects 2	Specifies the routing instance for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldBridgeDomain	jnxl2aldObjects 3	Specifies the bridge domain for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldLogicalRouter	jnxl2aldObjects 4	Specifies the logical system for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldMacLimit	jnxl2aldObjects 5	Specifies the maximum number of MAC addresses that can be learned by the routing instance.
jnxl2aldGbMacLimit	jnxl2aldObjects 6	Specifies the maximum number of MAC addresses that can be learned by the router.

## MAC Address Limit Traps

The enterprise-specific L2ALD MIB defines the following traps:

- jnxl2aldRoutingInstMacLimit, whose object identifier is {jnxl2aldNotification 1}, is generated when the number of MAC addresses for the given routing instance, jnxl2aldRoutingInst, exceeds the set limit. This trap contains the following objects: jnxl2aldLogicalRouter, jnxl2aldRoutingInst, jnxl2aldBridgeDomain, and jnxl2aldMacLimit.
- jnxl2aldInterfaceMacLimit, whose object identifier is {jnxl2aldNotification 2}, is generated when the number of MAC addresses for the given physical interface exceeds the set limit. This trap contains the following objects: jnxl2aldIntfLogicalRouter, jnxl2aldIntfRoutingInst, jnxl2aldIntfBridgeDomain, ifDescr, and jnxl2aldIntfMacLimit.
- jnxl2aldGlobalMacLimit, whose object identifier is {jnxl2aldNotification 3}, is generated when the MAC limit for the entire system exceeds the set limit.

## Chapter 62

# Interpreting the Enterprise-Specific Utility MIB

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

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

Each data has an arbitrary ASCII name, which is defined when the data is populated, and a timestamp that shows the last time when the data instance was modified. For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-util.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-util.txt).

This chapter contains the following topics:

- `jnxUtilCounter32Table` on page 569
- `jnxUtilCounter64Table` on page 570
- `jnxUtilIntegerTable` on page 570
- `jnxUtilUintTable` on page 570
- `jnxUtilStringTable` on page 571

## **jnxUtilCounter32Table**

---

`jnxUtilCounter32Table`, whose object ID is `{jnxUtilData 1}`, contains information on 32-bit counters.

Each `jnxUtilCounter32Entry` has the objects listed in Table 165 on page 570.

**Table 165: jnxUtilCounter32Entry**

Object	Object ID	Description
jnxUtilCounter32Name	jnxUtilCounter32Entry 1	Name assigned to the instance
jnxUtilCounter32Value	jnxUtilCounter32Entry 2	Value of the instance
jnxUtilCounter32Time	jnxUtilCounter32Entry 3	Time when the instance was last populated.

## jnxUtilCounter64Table

jnxUtilCounter64Table, whose object ID is {jnxUtilData 2}, contains information about 64-bit counters.

Each jnxUtilCounter64Entry has the objects listed in Table 166 on page 570.

**Table 166: jnxUtilCounter64Entry**

Object	Object ID	Description
jnxUtilCounter64Name	jnxUtilCounter64Entry 1	Name assigned to the instance
jnxUtilCounter64Value	jnxUtilCounter64Entry 2	Value of the instance
jnxUtilCounter64Time	jnxUtilCounter64Entry 3	Time when the instance was last populated.

## jnxUtilIntegerTable

jnxUtilIntegerTable, whose object ID is {jnxUtilData 3}, contains information about signed integer values.

Each jnxUtilIntegerEntry contains the objects listed in Table 167 on page 570.

**Table 167: jnxUtilIntegerEntry**

Object	Object ID	Description
jnxUtilIntegerName	jnxUtilIntegerEntry 1	Name assigned to the instance
jnxUtilIntegerValue	jnxUtilIntegerEntry 2	Value of the instance
jnxUtilIntegerTime	jnxUtilIntegerEntry 3	Time when the instance was last populated.

## jnxUtilUintTable

jnxUtilUintTable, whose object ID is {jnxUtilData 4}, contains information about unsigned integer values.



Each `jnxUtilUintEntry` has the objects listed in Table 168 on page 571.

**Table 168: `jnxUtilUintEntry`**

Object	Object ID	Description
<code>jnxUtilUintName</code>	<code>jnxUtilUintEntry 1</code>	Name assigned to the instance
<code>jnxUtilUintValue</code>	<code>jnxUtilUintEntry 2</code>	Value of the instance
<code>jnxUtilUintTime</code>	<code>jnxUtilUintEntry 3</code>	Time when the instance was last populated.

## **`jnxUtilStringTable`**

`jnxUtilStringTable`, whose object ID is `{jnxUtilData 5}`, contains information about octate strings.

Each `jnxUtilStringEntry` contains the objects listed in Table 169 on page 571.

**Table 169: `jnxUtilStringEntry`**

Object	Object ID	Description
<code>jnxUtilStringName</code>	<code>jnxUtilStringEntry 1</code>	Name assigned to the instance
<code>jnxUtilStringValue</code>	<code>jnxUtilStringEntry 2</code>	Value of the instance
<code>jnxUtilStringTime</code>	<code>jnxUtilStringEntry 31</code>	Time when the instance was last populated.



## Chapter 63

# Interpreting the Enterprise-Specific AAA Objects MIB

The enterprise-specific AAA Objects MIB, whose object ID is {jnxUserAAAMibRoot 1}, defines the objects pertaining to user authentication, authorization, and accounting.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-user-aaa.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-user-aaa.txt).

Object IDs for next branch nodes are as follows:

- jnxUserAAAGlobalStats—{jnxUserAAAObjects 1}
- jnxUserAAAAccessAuthStats—{jnxUserAAAObjects 2}
- jnxUserAAATrapVars—{jnxUserAAAObjects 3}

Object IDs for statistic counters related to access authentication are as follows:

- jnxTotalAuthenticationRequests—{jnxUserAAAGlobalStats 1}
- jnxTotalAuthenticationResponses—{jnxUserAAAGlobalStats 2}

This chapter contains the following topics:

- Text Conventions on page 573
- jnxUserAAASatTable on page 574
- jnxUserAAAServerName on page 574
- Access Authentication-Related Traps on page 574

## Text Conventions

JnxAuthenticateType—Defines the method to authenticate a user:

**Table 170: JnxAuthenticateType**

Method	Syntax Integer
radius—authentication via a radius server	(1)
local—local authentication	(2)

**Table 170: JnxAuthenticateType (continued)**

Method	Syntax Integer
ldap—authentication via a LDAP server	(3)

## jnxUserAAAStatTable

jnxUserAAAStatTable, whose object ID is {jnxUserAAAAccessAuthStats 1}, exposes the user authentication statistics listed in Table 171 on page 574.

**Table 171: jnxUserAAAStatTable**

Object	Object ID	Description
jnxUserAAAStatEntry	jnxUserAAAStatTable 1	Statistics entries collected for authentication.  Sequence of parameters: <ul style="list-style-type: none"> <li>■ jnxUserAAAStatAuthType</li> <li>■ jnxUserAAAStatRequestReceived</li> <li>■ jnxUserAAAStatAccessAccepted</li> <li>■ jnxUserAAAStatAccessRejected</li> </ul>
jnxUserAAAStatAuthType	jnxUserAAAStatEntry 1	Indicates the authentication type. This entry uniquely identifies the statistics counters related to its authentication.
jnxUserAAAStatRequestReceived	jnxUserAAAStatEntry 2	The number of the request received.
jnxUserAAAStatAccessAccepted	jnxUserAAAStatEntry 3	The number of the access granted. This entry is an aggregated statistic for this type of authentication.
jnxUserAAAStatAccessRejected	jnxUserAAAStatEntry 4	This number of the access request rejected. This entry is an aggregated statistic for this type of authentication.

## jnxUserAAAServerName

jnxUserAAAServerName, whose object ID is {jnxUserAAAAccessAuthStats 1}, specifies the server name that identifies the authentication server.

## Access Authentication-Related Traps

Table 172 on page 574 identifies access-authentication traps.

**Table 172: Access Authentication-Related Traps**

Object	Object ID	Description
jnxAccessAuthServiceUp	jnxUserAAANotifications 1	Access authentication trap to signify that the specified service has started

**Table 172: Access Authentication-Related Traps** *(continued)*

Object	Object ID	Description
jnxAccessAuthServiceDown	jnxUserAAANotifications 2	Access authentication trap to signify that the specified service has been stopped
jnxAccessAuthServerDisabled	jnxUserAAANotifications 3	Access authentication trap to signify that the external authentication server is not responding
jnxAccessAuthServerEnabled	jnxUserAAANotifications 4	Access authentication trap to signify that the external authentication server started responding again



## Chapter 64

# Interpreting the Enterprise-Specific Access Authentication Objects MIB

The enterprise-specific Access Authentication Objects MIB, whose object ID is {jnxJsAuth 1}, defines the objects that pertain to access authentication. Firewall and security features restrict the accessing of protected resources (ideally on different zones) behind a firewall based on their source IP and other credentials.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-auth.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-auth.txt).

This chapter contains the following topics:

- jnxJsFwAuthStats on page 577
- jnxJsAuthTrapVars on page 578
- jnxJsAuthNotifications on page 578

## jnxJsFwAuthStats

jnxJsFwAuthStats, whose object ID is {jnxJsAuthObjects 1}, exposes the firewall authentication statistics listed in Table 173 on page 577.

**Table 173: jnxJsFwAuthStats**

Object	Object ID	Description
jnxJsFwAuthNumPendingUsers	jnxJsFwAuthStats 1	Number of users currently waiting to be authenticated by the firewall user authentication mechanism
jnxJsFwAuthNumSuccUsers	jnxJsFwAuthStats 2	Number of users currently allowed access by the firewall user authentication mechanism
jnxJsFwAuthNumFailedUsers	jnxJsFwAuthStats 3	Number of users currently failed to be authenticated by the firewall user authentication mechanism
jnxJsFwAuthTotalUsers	jnxJsFwAuthStats 4	Total number of users that are accessing or attempting to access resources managed by the firewall user authentication mechanism

## jnxJsAuthTrapVars

jnxJsAuthTrapVars identifies access authentication traps variables listed in Table 174 on page 578.

**Table 174: jnxJsAuthTrapVars**

Object	Object ID	Description
jnxJsFwAuthUserName	jnxJsAuthTrapVars 1	Name of the user who is attempting to be authenticated or has been authenticated
jnxJsFwAuthServiceDesc	jnxJsAuthTrapVars 2	Service or application name that the authentication is performed for: Telnet, FTP, HTTP
jnxJsFwAuthReason	jnxJsAuthTrapVars 3	Reason for the trap being generated: authentication failure due to: timeout, invalid password, invalid username, and so on
jnxJsFwAuthClientIpAddr	jnxJsAuthTrapVars 4	Authentication client's IP address

## jnxJsAuthNotifications

jnxJsAuthNotifications, whose object ID is {jnxJsAuthMIB 0 }, identifies the user access authentication notifications listed in Table 175 on page 578 .

**Table 175: jnxJsAuthNotifications**

Object	Object ID	Description
jnxJsFwAuthFailure	jnxJsAuthNotifications 1	<p>A firewall user authentication status trap to signify whether a user using the pass-through firewall authentication mechanism has been rejected due to reason specified in the trap.</p> <ul style="list-style-type: none"> <li>■ jnxJsFwAuthUserName is the user.</li> <li>■ jnxClientIpAddress is the IP address the user came from.</li> <li>■ jnxJsFwAuthServiceDesc specifies the application by which the authentication was performed.</li> <li>■ jnxJsFwAuthReason indicates the reason for failure.</li> </ul>
jnxJsFwAuthServiceUp	jnxJsAuthNotifications 2	Firewall user authentication service has started.
jnxJsFwAuthServiceDown	jnxJsAuthNotifications 3	Firewall user authentication service has stopped.
jnxJsFwAuthCapacityExceeded	jnxJsAuthNotifications 4	<p>Firewall user authentication maximum capacity has been exceeded.</p> <p>jnxJsFwAuthTotalUsers indicates the total number of users being authenticated, and it has exceeds the maximum allowable users.</p>



## Chapter 65

# Interpreting the Enterprise-Specific DNS Objects MIB

The enterprise-specific DNS Objects MIB, `jnxJsDns`, whose object ID is `{jnxJsDnsRoot 1}`, provides collated statistics for the Domain Name System (DNS) proxy collected over all interfaces on which it is configured to serve.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-dns.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-dns.txt).

This chapter contains the following topic:

- `jnxJsDnsProxyDataObjects` on page 579

## `jnxJsDnsProxyDataObjects`

`jnxJsDnsProxyDataObjects`, whose object ID is `{jnxJsDns 1}`, displays the DNS query-related statistics listed in Table 176 on page 579.

**Table 176: `jnxJsDnsProxyDataObjects`**

Object	Object ID	Description
<code>jnxJsDNSProxyQueriesReceived</code>	<code>jnxJsDnsProxyDataObjects 1</code>	Total number of DNS queries received by the DNS proxy.
<code>jnxJsDnsProxyResponsesSent</code>	<code>jnxJsDnsProxyDataObjects 2</code>	Number of DNS queries answered sent by the DNS proxy. This includes DNS cache hits and misses that were answered.
<code>jnxJsDnsProxyQueriesForwarded</code>	<code>jnxJsDnsProxyDataObjects 3</code>	Number of DNS queries forwarded to other DNS servers. This is the number of queries that have been proxied due to cache misses.
<code>jnxJsDnsProxyNegativeResponses</code>	<code>jnxJsDnsProxyDataObjects 4</code>	Number of negative DNS query responses. This is the count of DNS queries for which the proxy could not obtain answers.
<code>jnxJsDnsProxyRetryRequests</code>	<code>jnxJsDnsProxyDataObjects 5</code>	Number of DNS retry queries that this proxy received.
<code>jnxJsDnsProxyPendingRequests</code>	<code>jnxJsDnsProxyDataObjects 6</code>	Number of DNS requests yet to be answered.
<code>jnxJsDnsProxyServerFailures</code>	<code>jnxJsDnsProxyDataObjects 7</code>	Number of DNS proxy failures.



## Chapter 66

# Interpreting the Enterprise-Specific IPSec Generic Flow Monitoring Object MIB

The enterprise-specific IPSec Generic Flow Monitoring Object MIB, whose object ID is {jnxIpSecMibRoot 1}, defines the objects used to monitor the entries pertaining to IPSec objects and the management of the IPSec VPN functionalities. This generic MIB models the standard, dynamic aspects of IPSec, including the counters and objects that are of management interest in a standard IPSec implementation.

This MIB module is based on the jnxIpSecMonitorMib. Building on the existing Internet Key Exchange (IKE) infrastructure, the security IKE implementation integrates the value-added features for the security products.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-flow-mon.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ipsec-flow-mon.txt).

This chapter contains the following topics:

- Branch Tree Objects on page 581
- Text Conventions on page 582
- Number of IKE Tunnels Currently Active on page 585
- IPSec Phase 1 IKE Tunnel Table on page 586
- IPSec Phase 2 IKE Tunnel Table on page 589
- IPSec Phase 2 Security Association Table on page 592

## Branch Tree Objects

---

The following branch tree objects are associated with the IPSec Generic Flow Monitoring Object MIB:

- jnxIpSecFlowMonNotifications {jnxIpSecFlowMonMIB 0}
- jnxIpSecFlowMonPhaseOne {jnxIpSecFlowMonMIB 1}
- jnxIpSecFlowMonPhaseTwo {jnxIpSecFlowMonMIB 2}

## Text Conventions

- **JnxIkePeerType**—The type of IPsec Phase 1 IKE peer identity. This identity is the local IKE identity to send in the exchange. The IKE peer may be identified by one of the ID types defined in IPsec DOI:

**Table 177: IKE Identity Type Text Conventions**

Type	Description	Syntax Integer
Unknown	Unknown type	(0)
idIpv4Addr	IPv4 address	(1)
idFqdn	Fully qualified domain name	(2)
idDn	Distinguished name	(3)
idUfqdn	User fully qualified domain name	(4)

- **JnxIkeNegoMode**—The IPsec Phase 1 IKE negotiation mode:

**Table 178: IKE Negotiation Mode Text Conventions**

Type	Description	Syntax Integer
Main mode	A six-message Phase 1 exchange that provides identity protection	(1)
Aggressive mode	A three-message Phase 1 exchange that does not provide identity protection	(2)

- **JnxIkeHashAlgo**—The hash algorithm used in IPsec Phase 1 IKE negotiations:

**Table 179: IKE Negotiations Hash Algorithms**

Hash Algorithm	Syntax Integer
md5	(1)
sha	(2)

- **JnxIkeAuthMethod**—The authentication method used in IPsec Phase 1 IKE negotiations:

**Table 180: IKE Authentication Method**

Method	Syntax Integer
preSharedKey	(1)
dssSignature	(2)
rsaSignature	(3)
rsaEncryption	(4)
revRsaEncryption	(5)
xauthPreSharedKey	(6)
xauthDssSignature	(7)
xauthRsaSignature	(8)
xauthRsaEncryption	(9)
xauthRevRsaEncryption	(10)

- **JnxIkePeerRole**—The role of the local endpoint in negotiating the IPsec Phase 1 IKE security association (SA). It can be either initiator or responder.

**Table 181: Role of Local Endpoint in Negotiations**

Role	Syntax Integer
Initiator	(1)
Responder	(2)

- **JnxIkeTunStateType**—State of the Phase 1 IKE negotiation:

**Table 182: State of Phase 1 IKE Negotiation**

State	Syntax Integer
Up	(1)
Down	(2)

- **JnxDiffHellmanGrp**—The Diffie-Hellman Group used in negotiations:

**Table 183: Diffie-Hellman Group in Negotiations**

Diffie Hellman Group	Description	Syntax Integer
Unknown	Unknown	(0)
modp768	768-bit MODP	(1)
modp1024	1024-bit MODP	(2)
modp1536	modp1536	(3)

- **JnxKeyType**—The type of key used by an IPSec Phase 2 Tunnel:

**Table 184: Key Used by IPSec Phase 2 Tunnel**

Key	Syntax Integer
Unknown	(0)
keylke	(1)
keyManual	(2)

- **JnxKeyType**—The encryption algorithm used in negotiations:

**Table 185: Encryption Algorithm in Negotiations**

Algorithm	Syntax Integer
espDes	(1)
esp3des	(2)
espNull	(3)
espAes128	(4)
espAes192	(5)
espAes256	(6)

- **JnxAuthAlgo**—The authentication algorithm used by an SA of an IPSec Phase 2 Tunnel:

**Table 186: Role of Local Endpoint in Negotiations**

Algorithm	Syntax Integer
Unknown	(0)
hmacMd5	(1)
hmacSha	(2)

- **JnxRemotePeerType**—The type of the remote peer gateway (endpoint). It can be one of the following two types:
  - Static (remote peer whose IP address is known beforehand)
  - Dynamic (remote peer whose IP address is not known beforehand)

**Table 187: Type of Remote Peer Gateway**

Gateway Type	Syntax Integer
Unknown	(0)
static	(1)
dynamic	(2)

- **JnxSpiType**—The type of the SPI associated with IPsec Phase 2 SAs. An unsigned 32-bit integer (256. . . 4294967295).
- **JnxSAType**—The SA type:

**Table 188: Role of Local Endpoint in Negotiations**

SA Type	Syntax Integer
Unknown	(0)
manual	(1)
dynamic	(2)

## Number of IKE Tunnels Currently Active



**NOTE:** The `jnxIkeNumOfTunnels` object is not supported in this release.

Table 189 on page 586 reports the number of IKE Tunnels currently active.

**Table 189: Number of IKE Tunnels Currently Active**

Object	Object ID	Description
jnxIkeNumOfTunnels	jnxIpSecFlowMonPhaseOne 1	Number of IKE Tunnels (Phase 1) actively negotiating between peers. The SA can be in either the up or down state. This attribute details the number of IKE tunnels in jnxIkeTunnelMonTable.

## IPSec Phase 1 IKE Tunnel Table

Table 190 on page 586 identifies objects listed in the IPSec Phase 1 IKE Tunnel Table.

Phase 1 is used to negotiate the parameter and key material required to establish an ISAKMP SA.

Phase 1 SA components include encryption algorithm, authentication, Diffie-Hellman group values, and authentication method, such as preshared keys or certificates.

**Table 190: IPSec Phase 1 IKE Tunnel Table**

Object	Object ID	Description
jnxIkeTunnelMonTable	jnxIpSecFlowMonPhaseOne 2	The IPSec Phase 1 IKE Tunnel Table. There is one entry in this table for each active IPSec Phase 1 IKE tunnel.



**Table 190: IPsec Phase 1 IKE Tunnel Table** (continued)

Object	Object ID	Description
jnxIkeTunnelMonEntry	jnxIkeTunnelMonTable 1	<p>Attributes associated with an active IPsec Phase 1 IKE tunnel.</p> <p>Sequence of attributes:</p> <ul style="list-style-type: none"> <li>■ jnxIkeTunMonRemoteGwAddrType</li> <li>■ jnxIkeTunMonRemoteGwAddr</li> <li>■ jnxIkeTunMonIndex</li> <li>■ jnxIkeTunMonLocalGwAddrType</li> <li>■ jnxIkeTunMonLocalGwAddr</li> <li>■ jnxIkeTunMonState</li> <li>■ jnxIkeTunMonInitiatorCookie</li> <li>■ jnxIkeTunMonResponderCookie</li> <li>■ jnxIkeTunMonLocalRole</li> <li>■ jnxIkeTunMonLocalIdType</li> <li>■ jnxIkeTunMonLocalIdValue</li> <li>■ jnxIkeTunMonLocalCertName</li> <li>■ jnxIkeTunMonRemotIdType</li> <li>■ jnxIkeTunMonRemotIdValue</li> <li>■ jnxIkeTunMonNegoMode</li> <li>■ jnxIkeTunMonDiffHellmanGrp (not supported in this release)</li> <li>■ jnxIkeTunMonEncryptAlgo</li> <li>■ jnxIkeTunMonHashAlgo</li> <li>■ jnxIkeTunMonAuthMethod</li> <li>■ jnxIkeTunMonLifeTime</li> <li>■ jnxIkeTunMonActiveTime</li> <li>■ jnxIkeTunMonInOctets</li> <li>■ jnxIkeTunMonInPkts</li> <li>■ jnxIkeTunMonOutOctets</li> <li>■ jnxIkeTunMonOutPkts</li> <li>■ jnxIkeTunMonXAuthUserId</li> <li>■ jnxIkeTunMonDPDDownCount</li> </ul>
jnxIkeTunMonRemoteGwAddrType	jnxIkeTunnelMonEntry 1	IP address type of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel
jnxJsFwAuthClientIpAddr	jnxJsAuthTrapVars 4	IP address of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel
jnxIkeTunMonIndex	jnxIkeTunnelMonEntry 3	Index number of IPsec Phase 1 IKE Tunnel Table. The index number begins at 1 and is incremented with each tunnel that is created. The value of this object will wrap at 2,147,483,647.
jnxIkeTunMonLocalGwAddr	jnxIkeTunnelMonEntry 4	IP address of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel
jnxIkeTunMonLocalGwAddrType	jnxIkeTunnelMonEntry 5	IP address type of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel

**Table 190: IPSec Phase 1 IKE Tunnel Table** (continued)

Object	Object ID	Description
jnxIkeTunMonState	jnxIkeTunnelMonEntry 6	State of IKE tunnel. It can be: <ul style="list-style-type: none"> <li>■ 1—up, negotiation completed</li> <li>■ 2—down, being negotiated</li> </ul>
jnxIkeTunMonInitiatorCookie	jnxIkeTunnelMonEntry 7	Cookie as generated by peer that initiated the IKE Phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunMonResponderCookie	jnxIkeTunnelMonEntry 8	Cookie as generated by peer responding to the IKE Phase 1 negotiation initiated by the remote peer. This cookie is carried in the ISAKMP header.
jnxIkeTunMonLocalRole	jnxIkeTunnelMonEntry 9	Role of local peer identity. The role of the local peer can be: <ul style="list-style-type: none"> <li>■ Initiator</li> <li>■ Responder</li> </ul>
jnxIkeTunMonLocalIdType	jnxIkeTunnelMonEntry 10	Type of local peer identity. The local peer can be identified by: <ul style="list-style-type: none"> <li>■ IP address</li> <li>■ Fully qualified domain name string</li> <li>■ Distinguished name string</li> </ul>
jnxIkeTunMonLocalIdValue	jnxIkeTunnelMonEntry 11	Value of local peer identity. <p>If the local peer type is an IP address, then this is the IP address used to identify the local peer.</p> <p>If the local peer type is a fully qualified domain name string, then this is the fully qualified domain name string of the local peer.</p> <p>If the local peer type is a distinguished name string, then this is the distinguished name string of the local peer.</p>
jnxIkeTunMonLocalCertName	jnxIkeTunnelMonEntry 12	Name of certificate used for authentication of the local tunnel endpoint. This object has some valid value only if the negotiated IKE authentication method is other than preshared key. If the IKE negotiation does not use a certificate-based authentication method, then the value of this object is a NULL string.
jnxIkeTunMonRemoteIdType	jnxIkeTunnelMonEntry 13	Type of remote peer identity. The remote peer can be identified by: <ul style="list-style-type: none"> <li>■ IP address</li> <li>■ Fully qualified domain name string</li> <li>■ Distinguished name string</li> </ul>

**Table 190: IPsec Phase 1 IKE Tunnel Table** (continued)

Object	Object ID	Description
jnxIkeTunMonRemoteldValue	jnxIkeTunnelMonEntry 14	Value of remote peer identity.  If the remote peer type is an IP address, then this is the IP address used to identify the remote peer.  If the remote peer type is a fully qualified domain name string, then this is the fully qualified domain name string of the remote peer.  If the remote peer type is a distinguished name string, then this is the distinguished name string of the remote peer.
jnxIkeTunMonNegoMode	jnxIkeTunnelMonEntry 15	Negotiation mode of IPsec Phase 1 IKE tunnel
<b>NOTE:</b> The jnxIkeTunMonDiffHellmanGrp object is not supported in this release.		
jnxIkeTunMonDiffHellmanGrp	jnxIkeTunnelMonEntry 16	Diffie-Hellman Group used in IPsec Phase 1 IKE negotiations
jnxIkeTunMonEncryptAlgo	jnxIkeTunnelMonEntry 17	Encryption algorithm used in IPsec Phase 1 IKE negotiations
jnxIkeTunMonHashAlgo	jnxIkeTunnelMonEntry 18	Hash algorithm used in IPsec Phase 1 IKE negotiations
jnxIkeTunMonAuthMethod	jnxIkeTunnelMonEntry 19	Authentication method used in IPsec Phase 1 IKE negotiations
jnxIkeTunMonLifeTime	jnxIkeTunnelMonEntry 20	Negotiated lifetime of IPsec Phase 1 IKE tunnel in seconds
jnxIkeTunMonActiveTime	jnxIkeTunnelMonEntry 21	Length of time IPsec Phase 1 IKE tunnel has been active in hundredths of seconds
jnxIkeTunMonInOctets	jnxIkeTunnelMonEntry 22	Total number of octets received by this IPsec Phase 1 IKE SA
jnxIkeTunMonInPkts	jnxIkeTunnelMonEntry 23	Total number of packets received by this IPsec Phase 1 IKE SA
jnxIkeTunMonOutOctets	jnxIkeTunnelMonEntry 24	Total number of octets sent by this IPsec Phase 1 IKE SA
jnxIkeTunMonOutPkts	jnxIkeTunnelMonEntry 25	Total number of packets sent by this IPsec Phase 1 IKE SA
jnxIkeTunMonXAuthUserId	jnxIkeTunnelMonEntry 26	Extended Authentication (XAuth) User Identifier. Identifies the user associated with this IPsec Phase 1 negotiation
jnxIkeTunMonDPDDownCount	jnxIkeTunnelMonEntry 27	Number of times that the remote peer is detected in a dead (or down) state

## IPsec Phase 2 IKE Tunnel Table

Table 191 on page 590 identifies objects listed in the IPsec Phase 2 IKE Tunnel Table.

During this phase, IKE negotiates IPsec SA parameters and setup, matching IPsec SA in the peers.

Phase 2 VPN includes tunnel peer connection, associated with a specific policy or a tunnel interface. Phase 2 SA components include encryption and authentication algorithms, proxy-IDs, and optional DH group values.

**Table 191: IPSec Phase 2 IKE Tunnel Table**

Object	Object ID	Description
<b>NOTE:</b> The jnxIpSecNumOfTunnels object is not supported in this release.		
jnxIpSecNumOfTunnels	jnxIpSecFlowMonPhaseTwo 1	Number of IPSec VPN tunnels. This attribute should report the number of IPSec VPN tunnels in jnxIpSecTunnelTable.
jnxIpSecTunnelMonTable	jnxIpSecFlowMonPhaseTwo 2	The IPSec Phase 2 Tunnel Table. There is one entry in this table for each active IPSec Phase 2 tunnel. If the tunnel is terminated, then the entry is no longer available after the table has been refreshed.
jnxIpSecTunnelMonEntry	jnxIpSecTunnelMonTable 1	<p>Each entry contains the attributes associated with an active IPSec Phase 2 tunnel.</p> <p>Sequence of attributes:</p> <ul style="list-style-type: none"> <li>■ jnxIpSecTunMonRemoteGwAddrType</li> <li>■ jnxIpSecTunMonRemoteGwAddr</li> <li>■ jnxIpSecTunMonIndex</li> <li>■ jnxIpSecTunMonLocalGwAddrType</li> <li>■ jnxIpSecTunMonLocalGwAddr</li> <li>■ jnxIpSecTunMonLocalProxyId</li> <li>■ jnxIpSecTunMonRemoteProxyId</li> <li>■ jnxIpSecTunMonKeyType</li> <li>■ jnxIpSecTunMonRemotePeerType</li> <li>■ jnxIpSecTunMonOutEncryptedBytes</li> <li>■ jnxIpSecTunMonOutEncryptedPkts</li> <li>■ jnxIpSecTunMonInDecryptedBytes</li> <li>■ jnxIpSecTunMonInDecryptedPkts</li> <li>■ jnxIpSecTunMonAHInBytes</li> <li>■ jnxIpSecTunMonAHInPkts</li> <li>■ jnxIpSecTunMonAHOutBytes</li> <li>■ jnxIpSecTunMonAHOutPkts</li> <li>■ jnxIpSecTunMonReplayDropPkts</li> <li>■ jnxIpSecTunMonAhAuthFails</li> <li>■ jnxIpSecTunMonDecryptFails</li> <li>■ jnxIpSecTunMonBadHeaders</li> <li>■ jnxIpSecTunMonBadTrailers</li> <li>■ jnxIkeTunMonOutOctets</li> <li>■ jnxIpSecTunMonDroppedPkts (not supported in this release)</li> </ul>
jnxIpSecTunMonRemoteGwAddrType	jnxIpSecTunnelMonEntry 1	IP address type of remote gateway (endpoint) for the IPSec Phase 2 tunnel

**Table 191: IPSec Phase 2 IKE Tunnel Table** (continued)

Object	Object ID	Description
jnxIpSecTunMonRemoteGwAddr	jnxIpSecTunnelMonEntry 2	IP address of remote gateway (endpoint) for the IPSec Phase 2 tunnel
jnxIpSecTunMonIndex	jnxIpSecTunnelMonEntry 3	Index number of IPSec Phase 2 Tunnel Table. The index number begins at 1 and is incremented with each tunnel that is created. The value of this object will wrap at 2,147,483,647.
jnxIpSecTunMonLocalGwAddrType	jnxIpSecTunnelMonEntry 4	IP address type of local gateway (endpoint) for the IPSec Phase 2 tunnel
jnxIpSecTunMonLocalGwAddr	jnxIpSecTunnelMonEntry 5	IP address of local gateway (endpoint) for the IPSec Phase 2 tunnel
jnxIpSecTunMonLocalProxyId	jnxIpSecTunnelMonEntry 6	Identifier for local end
jnxIpSecTunMonRemoteProxyId	jnxIpSecTunnelMonEntry 7	Identifier for remote end
jnxIpSecTunMonKeyType	jnxIpSecTunnelMonEntry 8	Type of key used by IPSec Phase 2 tunnel. It can be one of the following two types: <ul style="list-style-type: none"> <li>■ IKE-negotiated</li> <li>■ Manually installed</li> </ul>
jnxIpSecTunMonRemotePeerType	jnxIpSecTunnelMonEntry 9	Type of the remote peer gateway (endpoint). It can be one of the following two types: <ul style="list-style-type: none"> <li>■ Static (remote peer whose IP address is known beforehand)</li> <li>■ Dynamic (remote peer whose IP address is not known beforehand)</li> </ul>
jnxIpSecTunMonOutEncryptedBytes	jnxIpSecTunnelMonEntry 10	Number of bytes encrypted by this Phase 2 tunnel
jnxIpSecTunMonOutEncryptedPkts	jnxIpSecTunnelMonEntry 11	Number of packets encrypted by this Phase 2 tunnel
jnxIpSecTunMonInDecryptedBytes	jnxIpSecTunnelMonEntry 12	Number of bytes decrypted by this Phase 2 tunnel
jnxIpSecTunMonInDecryptedPkts	jnxIpSecTunnelMonEntry 13	Number of packets decrypted by this Phase 2 tunnel
jnxIpSecTunMonAHInBytes	jnxIpSecTunnelMonEntry 14	Number of incoming bytes authenticated using AH by this Phase 2 tunnel
jnxIpSecTunMonAHInPkts	jnxIpSecTunnelMonEntry 15	Number of incoming packets authenticated using AH by this Phase 2 tunnel
jnxIpSecTunMonAHOuBytes	jnxIpSecTunnelMonEntry 16	Number of outgoing bytes applied AH by this Phase 2 tunnel
jnxIpSecTunMonAHOuPkts	jnxIpSecTunnelMonEntry 17	Number of outgoing packets applied AH by this Phase 2 tunnel.
jnxIpSecTunMonReplayDropPkts	jnxIpSecTunnelMonEntry 18	Number of packets dropped by this Phase 2 tunnel due to antireplay check failure

**Table 191: IPSec Phase 2 IKE Tunnel Table** *(continued)*

Object	Object ID	Description
jnxIpSecTunMonAhAuthFails	jnxIpSecTunnelMonEntry 19	Number of packets received by this Phase 2 tunnel that failed AH authentication
jnxIpSecTunMonEspAuthFails	jnxIpSecTunnelMonEntry 20	Number of packets received by this Phase 2 tunnel that failed ESP authentication
jnxIpSecTunMonDecryptFails	jnxIpSecTunnelMonEntry 21	Number of packets received by this Phase 2 tunnel that failed decryption
jnxIpSecTunMonBadHeaders	jnxIpSecTunnelMonEntry 22	Number of packets received by this Phase 2 tunnel that failed due to bad headers
jnxIpSecTunMonBadTrailers	jnxIpSecTunnelMonEntry 23	Number of packets received by this Phase 2 tunnel that failed due to bad ESP trailers
<b>NOTE:</b> The jnxIpSecTunMonDroppedPkts object is not supported in this release.		
jnxIpSecTunMonDroppedPkts	jnxIpSecTunnelMonEntry 26	Total number of dropped packets for this Phase 2 tunnel

## IPSec Phase 2 Security Association Table

jnxIpSecSaMonTable, whose object ID is {jnxIpSecFlowMonPhaseTwo 3}, identifies the objects listed in Table 192 on page 593. The IPSec Phase 2 Security Association table identifies the structure (in terms of component SAs) of each active Phase 2 IPSec tunnel. This table contains an entry for each active and expiring SA and maps each entry in the active Phase 2 tunnel table (ipSecTunTable) into a number of entries in this table.

SA contains the information negotiated by IKE. The SA is like a contract laying out the rules of the VPN connection for the duration of the SA. An SA is assigned a 32-bit number that, when used in conjunction with the destination IP address, uniquely identifies the SA. This number is called the Security Parameters Index (SPI).

IPSec SAs are unidirectional and are unique in each security protocol. A set of SAs is needed for a protected data pipe, one per direction per protocol.

**Table 192: IPsec Phase 2 Security Association Table**

Object	Object ID	Description
jnxIpSecSaMonEntry	jnxIpSecSaMonTable 1	<p>Each entry contains the attributes associated with active and expiring IPsec Phase 2 SAs.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ jnxIpSecSaMonIndex</li> <li>■ jnxIpSecSaMonProtocol</li> <li>■ jnxIpSecSaMonInSpi</li> <li>■ jnxIpSecSaMonOutSpi</li> <li>■ jnxIpSecSaMonType</li> <li>■ jnxIpSecSaMonEncapMode</li> <li>■ jnxIpSecSaMonLifeSize</li> <li>■ jnxIpSecSaMonLifeTime</li> <li>■ jnxIpSecSaMonActiveTime</li> <li>■ jnxIpSecSaMonLifeSizeThreshold (not supported in this release)</li> <li>■ jnxIpSecSaMonLifeTimeThreshold</li> <li>■ jnxIpSecSaMonEncryptAlgo</li> <li>■ jnxIpSecSaMonAuthAlgo</li> <li>■ jnxIpSecSaMonState</li> </ul>
jnxIpSecSaMonIndex	jnxIpSecSaMonEntry 1	Index number, in the context of the IPsec tunnel ipSecTunIndex, of the SA represented by this table entry. The index number begins at 1 and is incremented with each SPI associated with an IPsec Phase 2 tunnel. The value of this object will wrap at 65535.
jnxIpSecSaMonProtocol	jnxIpSecSaMonEntry 2	Index number that represents the security protocol (AH, ESP or IPComp) for which this SA was set up
jnxIpSecSaMonInSpi	jnxIpSecSaMonEntry 3	Value of the incoming SPI
jnxIpSecSaMonOutSpi	jnxIpSecSaMonEntry 4	Value of the outgoing SPI
jnxIpSecSaMonType	jnxIpSecSaMonEntry 5	Types of SAs that can be either manual or dynamic
jnxIpSecSaMonEncapMode	jnxIpSecSaMonEntry 6	Encapsulation mode used by an IPsec Phase 2 tunnel
jnxIpSecSaMonLifeSize	jnxIpSecSaMonEntry 7	Negotiated lifesize of the IPsec Phase 2 tunnel in kilobytes
jnxIpSecSaMonLifeTime	jnxIpSecSaMonEntry 8	Negotiated lifetime of the IPsec Phase 2 tunnel in seconds
jnxIpSecSaMonActiveTime	jnxIpSecSaMonEntry 9	Length of time the IPsec Phase 2 tunnel has been active in hundredths of seconds
<b>NOTE:</b> The jnxIpSecSaMonLifeSizeThreshold object is not supported in this release.		
jnxIpSecSaMonLifeSizeThreshold	jnxIpSecSaMonEntry 10	SA lifesize refresh threshold in kilobytes
jnxIpSecSaMonLifeTimeThreshold	jnxIpSecSaMonEntry 11	SA lifetime refresh threshold in seconds

**Table 192: IPSec Phase 2 Security Association Table** (*continued*)

Object	Object ID	Description
jnxIpSecSaMonEncryptAlgo	jnxIpSecSaMonEntry 12	Encryption algorithm used to encrypt the packets that can be either <b>es-cbc</b> or <b>3des-cbc</b>
jnxIpSecSaMonAuthAlgo	jnxIpSecSaMonEntry 13	Algorithm used for authentication of packets that can be <b>hmac-md5-96</b> or <b>hmac-sha1-96</b>
jnxIpSecSaMonState	jnxIpSecSaMonEntry 14	This column represents the status of the SA represented by this table entry. If the status of the SA is <b>active</b> , the SA is ready for active use. The status <b>expiring</b> represents any of the various states that the SA transitions through before being purged.



## Chapter 67

# Interpreting the Enterprise-Specific IPsec VPN Objects MIB

The enterprise-specific IPsec VPN Objects MIB, `jnxJsIpSecVpnMib`, whose object ID is `{jnxJsIpSecVpn 1}`, defines the object used to monitor the entries pertaining to IPsec objects and the management of the IPsec VPN functionalities for Juniper Networks security product lines. This MIB models IPsec attributes specific to the appropriate Juniper Networks implementation.

This MIB module extends the Juniper Networks common IPsec flow monitoring MIB. Building on the existing common infrastructure, the security implementation integrates the value-added features for the security products.

Related IPsec VPN Objects MIBs include:

- `jnxJsIpSecVpnNotifications` `{jnxJsIpSecVpnMib 0}`
- `jnxJsIpSecVpnPhaseOne` `{jnxJsIpSecVpnMib 1}`
- `jnxJsIpSecVpnPhaseTwo` `{jnxJsIpSecVpnMib 2}`

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-ipsec-vpn.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-ipsec-vpn.txt).

This chapter contains the following topics:

- Text Conventions on page 595
- `jnxJsIpSecTunnelTable` on page 596

## Text Conventions

`JnxJsIpSecVpnType`—Type of the remote peer gateway (endpoint):

**Table 193: JnxJsIpSecVpnType**

Type	Description	Syntax Integer
<code>policyBased</code>	Tunnels require a policy with action <code>tunnel</code> to trigger IPsec VPN. The device receives traffic and matches it with policy that has action <code>tunnel</code> , then performs the encryption/decryption and authentication options negotiated for this VPN Phase 2 negotiation.	(1)

**Table 193: JnxJslpSecVpnType (continued)**

Type	Description	Syntax Integer
routeBased	Requires a tunnel interface to a route directing traffic to protected networks to exit the system using that tunnel interface. The tunnel interface is bound to a Phase 2 VPN configuration that specifies all the tunnel parameters.	(2)

## jnxJslpSecTunnelTable

jnxJslpSecTunnelTable, whose object ID is {jnxJslpSecVpnPhaseTwo 1}, is the IPsec Phase 2 Tunnel Table, with objects listed in Table 194 on page 596. There is one entry for each active IPsec Phase 2 tunnel. If the tunnel is terminated, then the entry is no longer available after the table has been refreshed.

This table augments jnxIpSecTunnelMonTable in Table 191 on page 590.

**Table 194: jnxJslpSecTunnelTable**

Object	Object ID	Description
jnxJslpSecTunnelEntry	jnxJslpSecTunnelTable 1	Each entry contains the attributes associated with an active IPsec Phase 2 tunnel.  Sequence of parameters: <ul style="list-style-type: none"> <li>■ jnxJslpSecTunPolicyName</li> <li>■ jnxJslpSecVpnTunType</li> <li>■ jnxJslpSecTunCfgMonState</li> <li>■ jnxJslpSecTunState</li> </ul>
jnxJslpSecTunPolicyName	jnxJslpSecTunnelEntry 1	Policy name associated with this tunnel if the IPsec VPN is policy-based. If the IPsec VPN is not policy-based, this attribute is not applicable.
jnxJslpSecVpnTunType	jnxJslpSecTunnelEntry 2	Attribute to indicate whether the IPsec VPN tunnel is policy-based or route-based.
jnxJslpSecTunCfgMonState	jnxJslpSecTunnelEntry 3	According to user configuration, whether to monitor the IPsec tunnel to be alive or not: <ul style="list-style-type: none"> <li>■ disable—(1)</li> <li>■ enable—(2)</li> </ul>

**Table 194: jnxJslpSecTunnelTable** (continued)

Object	Object ID	Description
jnxJslpSecTunState	jnxJslpSecTunnelEntry 4	<p>Attribute to indicate whether the IPsec tunnel is up or down, determined by ICMP ping if jnxJslpSecTunCfgMonState is enabled:</p> <ul style="list-style-type: none"> <li>■ up—</li> <li>■ down—(2): VPN monitor detects the tunnel is down.</li> <li>■ vpnMonitoringDisabled—(3): User has disabled VPN tunnel monitoring.</li> </ul>



## Chapter 68

# Interpreting the Enterprise-Specific Network Address Translation Objects MIB

The enterprise-specific Network Address Translation (NAT) Objects MIB, `jnxJsNatMIB`, whose object ID is `{jnxJsNAT 1}`, defines the objects that are used to monitor NAT attributes.

Related NAT Objects MIB include:

- `jnxJsNatNotifications` `{jnxJsNatMIB 0}`
- `jnxJsNatObjects` `{jnxJsNatMIB 1}`
- `jnxJsNatTrapVars` `{jnxJsNatMIB 2}`

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-nat.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-nat.txt).

This chapter contains the following topics:

- Source NAT Table on page 599
- `jnxJsNatIfSrcPoolPortTable` on page 601
- NAT Trap Definitions on page 601

## Source NAT Table

Table 195 on page 599 identifies objects in the Source NAT Table.

**Table 195: Source NAT Table**

Object	Object ID	Description
<code>jnxJsSrcNatNumOfEntries</code>	<code>jnxJsNatObjects 1</code>	Total number of dynamic addresses being translated.

**Table 195: Source NAT Table** *(continued)*

Object	Object ID	Description
jnxJsSrcNatTable	jnxJsNatObjects 2	<p>Table that exposes the source NAT attributes of the translated addresses.</p> <p>When performing source IP address translation, the device translates the original source IP address or port number to a different one. The resource, address source pools, provide the security device with a supply of addresses from which to draw when performing source NAT.</p> <p>The security device has the following types of source pools:</p> <ul style="list-style-type: none"> <li>■ Source pool with Port Address Translation (PAT)</li> <li>■ Source pool without PAT</li> <li>■ Static source pool</li> </ul> <p>This table contains information on source IP address translation only.</p>
jnxJsSrcNatEntry	jnxJsSrcNatTable 1	<p>Source NAT address entries. This object is indexed by the address pool table and the address allocated.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ jnxJsNatSrcIpPoolName</li> <li>■ jnxJsNatSrcGlobalAddr</li> <li>■ jnxJsNatSrcPortPoolType</li> <li>■ jnxJsNatSrcNumOfPortInuse</li> <li>■ jnxJsNatSrcNumOfSessions</li> <li>■ jnxJsNatSrcAssociatedIf</li> </ul>
jnxJsNatSrcIpPoolName	jnxJsSrcNatEntry 1	Address pool from which the translated address is allocated.
jnxJsNatSrcGlobalAddr	jnxJsSrcNatEntry 2	Name of dynamic source IP address allocated from the address pool used in the NAT translation.
jnxJsNatSrcPortPoolType	jnxJsSrcNatEntry 3	<p>Source NAT can do address translation with or without PAT. The source port pool type indicates whether the address translation is done with PAT, without PAT, or as a static translation:</p> <ul style="list-style-type: none"> <li>■ <b>withPAT</b>—The security device translates both source IP address and port number of the packets.</li> <li>■ <b>withoutPAT</b>—The device performs source NAT for the IP address without performing PAT for the source port number.</li> <li>■ <b>static</b>—One range of IP addresses is statically mapped one-to-one to a shifted range of IP addresses.</li> </ul>
jnxJsNatSrcNumOfPortInuse	jnxJsSrcNatEntry 4	<p>Number of ports in use for this NAT address entry.</p> <p>This attribute is applicable to only NAT translation with PAT.</p>
jnxJsNatSrcNumOfSessions	jnxJsSrcNatEntry 5	<p>Number of sessions in use for this NAT address entry.</p> <p>This attribute is applicable to only NAT translation without PAT.</p>

**Table 195: Source NAT Table** (continued)

Object	Object ID	Description
jnxJsNatSrcAssociatedIf	jnxJsNatEntry 6	Index of interfaces associated with this NAT address entry.  For each interface, the value is a unique value, greater than zero.

## jnxJsNatIfSrcPoolPortTable

jnxJsNatIfSrcPoolPortTable, whose object ID is jnxJsNatObjects 3, monitors the port usage of the NAT interface source IP address pool by displaying information about the objects listed in Table 196 on page 601.

The interface source pool is predefined. This source pool is referenced in a policy in which it is configured. The security device translates the source IP address to the address of the egress interface for the traffic, matching a policy that references the interface source pool. The security device always applies PAT for the interface source pool.

**Table 196: jnxJsNatIfSrcPoolPortTable**

Object	Object ID	Description
jnxJsNatIfSrcPoolPortEntry	jnxJsNatIfSrcPoolPortTable 1	Source NAT address entries. This object is indexed by the address pool table and the address.  Sequence of parameters: <ul style="list-style-type: none"> <li>■ jnxJsNatIfSrcPoolIndex</li> <li>■ jnxJsNatIfSrcPoolTotalSinglePorts</li> <li>■ jnxJsNatIfSrcPoolAllocSinglePorts</li> <li>■ jnxJsNatIfSrcPoolTotalTwinPorts</li> <li>■ jnxJsNatIfSrcPoolAllocTwinPorts</li> </ul>
jnxJsNatIfSrcPoolIndex	jnxJsNatIfSrcPoolPortEntry 1	Index number of the port pool of this address pool.
jnxJsNatIfSrcPoolTotalSinglePorts	jnxJsNatIfSrcPoolPortEntry 2	Total number of single ports in a port pool.
jnxJsNatIfSrcPoolAllocSinglePorts	jnxJsNatIfSrcPoolPortEntry 3	Number of single ports in a port pool allocated or in use.
jnxJsNatIfSrcPoolTotalTwinPorts	jnxJsNatIfSrcPoolPortEntry 4	Total number of twin ports in a port pool.
jnxJsNatIfSrcPoolAllocTwinPorts	jnxJsNatIfSrcPoolPortEntry 5	Number of twin ports in a port pool allocated or in use.

## NAT Trap Definitions

Table 197 on page 602 lists NAT trap definition objects.

**Table 197: NAT Trap Definitions**

Object	Object ID	Description
jnxJsNatAddrPoolThresholdStatus	jnxJsNatNotifications 1	<p>NAT address pool utilization threshold status trap to signify that the address pool utilization either exceeds a certain percentage or is clear of that percentage.</p> <p>jnxJsNatSrcIpPoolName is the name of the resource pool.</p> <p>jnxJsNatAddrPoolUtil is the percentage of utilization of the address pool.</p>
jnxJsNatAddrPoolUtil	jnxJsNatTrapVars 1	Dynamic address pool utilization expressed as a percentage.



Chapter 69

Interpreting the Enterprise-Specific Policy Objects MIB

The enterprise-specific Policy Objects MIB, `jnxJsSecPolicyMIB`, whose object ID is `{jnxJsPolicies 1}`, defines the MIB for policy monitoring.

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining one or more kinds of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

The Juniper Networks security device enforces the security policies rules for the transit traffic in terms of which traffic can pass through the firewall and the actions taken on the traffic as it passes through the firewall.

- Related MIB objects include the following:
- `jnxJsPolicyNotifications`—`{jnxJsSecPolicyMIB 0}`
  - `jnxJsPolicyObjects`—`{jnxJsSecPolicyMIB 1}`
  - `jnxJsPolicyTrapVars`—`{jnxJsSecPolicyMIB 2}`

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-policy.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-policy.txt).

- This chapter contains the following topics:
- Security Policy Table on page 603
  - `jnxJsPolicyStatsTable` on page 605

Security Policy Table

Table 198 on page 603 describes the objects in the Security Policy Table.

Table 198: Security Policy Table

Object	Object ID	Description
<code>jnxJsPolicyNumber</code>	<code>jnxJsPolicyObjects 1</code>	Number of policies (regardless of their current state) present on this system.

**Table 198: Security Policy Table** *(continued)*

Object	Object ID	Description
jnxJsPolicyTable	jnxJsPolicyObjects 2	<p>Exposes the security policy entries. Security devices and routers provide a network boundary with a single point of entry and exit, which allows the screening and directing of traffic through the implementation of access policies. The access policies can permit, deny, encrypt, authenticate, prioritize, schedule, and monitor the traffic flow through the firewall.</p> <p>This table lists entries of policy. The number of policies are given by <code>jnxJsPolicyNumber</code>.</p>
jnxJsPolicyEntry	jnxJsPolicyTable 1	<p>An entry contains a security policy.</p> <p>Indexes:</p> <ul style="list-style-type: none"> <li>■ <code>nxJsPolicyFromZone</code></li> <li>■ <code>jnxJsPolicyToZone</code></li> <li>■ <code>jnxJsPolicyName</code></li> </ul> <p>Security policies are configured under FromZone and ToZone directions. Under a specific zone direction, each security policy contains a name, match-criteria, action, and other options.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ <code>jnxJsPolicyFromZone</code></li> <li>■ <code>jnxJsPolicyToZone</code></li> <li>■ <code>jnxJsPolicyName</code></li> <li>■ <code>jnxJsPolicySequenceNumber</code></li> <li>■ <code>jnxJsPolicyAction</code></li> <li>■ <code>jnxJsPolicyScheduler</code></li> <li>■ <code>jnxJsPolicyState</code></li> <li>■ <code>jnxJsPolicyStatsAvailability</code></li> <li>■ <code>jnxJsPolicyPerSecBytesThreshold</code></li> <li>■ <code>jnxJsPolicyPerMinKbytesThreshold</code></li> </ul>
jnxJsPolicyFromZone	jnxJsPolicyEntry 1	FromZone name
jnxJsPolicyToZone	jnxJsPolicyEntry 2	ToZone name
jnxJsPolicyName	jnxJsPolicyEntry 3	Name of the policy defined. The name consists of up to 256 ASCII characters and uniquely identifies the policy entry.

**Table 198: Security Policy Table** *(continued)*

Object	Object ID	Description
jnxJsPolicySequenceNumber	jnxJsPolicyEntry 4	Indication of the policy sequence order of the policy within a specific FromZone and ToZone pair. Policies are matched in a sequence in which the ordering is specified by this number.
jnxJsPolicyAction	jnxJsPolicyEntry 5	Indication of the actions performed when the criteria are matched  The actions <b>permit</b> , <b>reject</b> , and <b>deny</b> are user-configured policies.
jnxJsPolicyScheduler	jnxJsPolicyEntry 6	Name of the schedule attached to this policy. Certain schedules have a specified duration that may affect the status of the policy.
jnxJsPolicyState	jnxJsPolicyEntry 7	State of this policy: active, inactive, or unavailable.  The state can be affected by the scheduler if the scheduler has a specified duration.
jnxJsPolicyStatsAvailability	jnxJsPolicyEntry 8	Indication of whether the statistics counters are available and are actively updated. If available, a matching <b>jnxJsPolicyStatsEntry</b> exists for the policy.
jnxJsPolicyPerSecBytesThreshold	jnxJsPolicyEntry 9	Indication of the threshold value of bytes per second
jnxJsPolicyPerMinKbytesThreshold	jnxJsPolicyEntry 10	Indication of the threshold value of kbyte per minute

## jnxJsPolicyStatsTable

**jnxJsPolicyStatsTable**, whose object ID is {**jnxJsPolicyObjects 3**}, exposes the security policy statistics entries listed in Table 199 on page 606. These statistics can be enabled and disabled by configuration on a per policy basis.

**Table 199: jnxJsPolicyStatsTable**

Object	Object ID	Description
jnxJsPolicyStatsEntry	jnxJsPolicyStatsTable 1	<p>Contains security policy statistics.</p> <p>Indexes:</p> <ul style="list-style-type: none"> <li>■ jnxJsPolicyFromZone</li> <li>■ jnxJsPolicyToZone</li> <li>■ jnxJsPolicyName</li> </ul> <p>Security policies are configured under FromZone and ToZone direction. Under a specific zone direction, each security policy contains name, match-criteria, action, and other options.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ jnxJsPolicyStatsCreationTime</li> <li>■ jnxJsPolicyStatsInputBytes</li> <li>■ jnxJsPolicyStatsInputByteRate</li> <li>■ jnxJsPolicyStatsOutputBytes</li> <li>■ jnxJsPolicyStatsOutputByteRate</li> <li>■ jnxJsPolicyStatsInputPackets</li> <li>■ jnxJsPolicyStatsInputPacketRate</li> <li>■ jnxJsPolicyStatsOutputPackets</li> <li>■ jnxJsPolicyStatsOutputPacketRate</li> <li>■ jnxJsPolicyStatsNumSessions</li> <li>■ jnxJsPolicyStatsSessionRate</li> <li>■ jnxJsPolicyStatsSessionDeleted</li> <li>■ jnxJsPolicyStatsLookups</li> <li>■ jnxJsPolicyStatsCountAlarm</li> </ul>
jnxJsPolicyStatsCreationTime	jnxJsPolicyStatsEntry 1	<p>Creation timestamp of the policy statistics entry. The timestamp is modified during the creation and deletion of the policy statistics entry. When the timestamp changes, the policy entry statistics entry is assumed to be a new statistics entry and not associated with a previous statistic entry of the same indices.</p>
jnxJsPolicyStatsInputBytes	jnxJsPolicyStatsEntry 2	<p>Number of input bytes that enter the firewall through this policy</p>
jnxJsPolicyStatsInputByteRate	jnxJsPolicyStatsEntry 3	<p>Number of input bytes per second or the rate that enters the firewall through this policy</p>
jnxJsPolicyStatsOutputBytes	jnxJsPolicyStatsEntry 4	<p>Number of output bytes associated with this policy</p>

**Table 199: jnxJsPolicyStatsTable** (continued)

Object	Object ID	Description
jnxJsPolicyStatsOutputByteRate	jnxJsPolicyStatsEntry 5	Number of output bytes per second or the rate associated with this policy
jnxJsPolicyStatsInputPackets	jnxJsPolicyStatsEntry 6	Number of input packets that enter the firewall through this policy
jnxJsPolicyStatsInputPacketRate	jnxJsPolicyStatsEntry 7	Number of input packets per second or the input packet rate of the firewall through this policy
jnxJsPolicyStatsOutputPackets	jnxJsPolicyStatsEntry 8	Number of output packets associated with this policy
jnxJsPolicyStatsOutputPacketRate	jnxJsPolicyStatsEntry 9	Number of output packets per second or the rate associated with this policy
jnxJsPolicyStatsNumSessions	jnxJsPolicyStatsEntry 10	Number of sessions associated with this policy
jnxJsPolicyStatsSessionRate	jnxJsPolicyStatsEntry 11	Rate of the sessions associated with this policy
jnxJsPolicyStatsSessionDeleted	jnxJsPolicyStatsEntry 12	Number of sessions associated with this policy
jnxJsPolicyStatsLookups	jnxJsPolicyStatsEntry 13	Number of policy lookups performed
jnxJsPolicyStatsCountAlarm	jnxJsPolicyStatsEntry 14	Number of alarms counted when the traffic exceeds a certain threshold configuration



## Chapter 70

# Interpreting the Enterprise-Specific Security Interface Extension Objects MIB

The enterprise-specific Security Interface Extension Objects MIB, `jnxJsIfMIB`, whose object ID is `{jnxJsIf 1}`, defines the object that are used to monitor the entries in the interfaces that pertain to the security management of the interface.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-if-ext.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-if-ext.txt).

This chapter contains the following topic:

- `jnxJsIfMonTable` on page 609

## **`jnxJsIfMonTable`**

---

`jnxJsIfMonTable`, whose object ID is `{jnxJsIfExtension 1}`, extends the interface entries on a particular interface to support the security related-objects listed in Table 200 on page 610. The table is indexed by `ifIndex`.

**Table 200: jnxJsIfMonTable**

Object	Object ID	Description
jnxJsIfMonEntry	jnxJsIfMonTable 1	<p>Object related to interface monitoring</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ jnxJsIfMonInIcmp</li> <li>■ jnxJsIfMonInSelf</li> <li>■ jnxJsIfMonInVpn</li> <li>■ jnxJsIfMonInPolicyPermit</li> <li>■ jnxJsIfMonOutPolicyPermit</li> <li>■ jnxJsIfMonConn</li> <li>■ jnxJsIfMonInMcast</li> <li>■ jnxJsIfMonOutMcast</li> <li>■ jnxJsIfMonPolicyDeny</li> <li>■ jnxJsIfMonNoGateParent</li> <li>■ jnxJsIfMonTcpProxyDrop</li> <li>■ jnxJsIfMonNoDip</li> <li>■ jnxJsIfMonNoNspTunnel</li> <li>■ jnxJsIfMonNoNatCon</li> <li>■ jnxJsIfMonInvalidZone</li> <li>■ jnxJsIfMonIpClsFail</li> <li>■ jnxJsIfMonAuthDrop</li> <li>■ jnxJsIfMonMultiUserAuthDrop</li> <li>■ jnxJsIfMonLoopMultiDipDrop</li> <li>■ jnxJsIfMonAddrSpoof</li> <li>■ jnxJsIfMonLpDrop</li> <li>■ jnxJsIfMonNullZone</li> <li>■ jnxJsIfMonNoGate</li> <li>■ jnxJsIfMonNoMinorSess</li> <li>■ jnxJsIfMonNvecErr</li> <li>■ jnxJsIfMonTcpSeq</li> <li>■ jnxJsIfMonIllegalPak</li> <li>■ jnxJsIfMonNoRoute</li> <li>■ jnxJsIfMonAuthFail</li> <li>■ jnxJsIfMonSalnactive</li> <li>■ jnxJsIfMonNoSa</li> <li>■ jnxJsIfMonSelfPktDrop</li> </ul>
jnxJsIfMonInIcmp	jnxJsIfMonEntry 1	ICMP packets received
jnxJsIfMonInSelf	jnxJsIfMonEntry 2	Self packets received
jnxJsIfMonInVpn	jnxJsIfMonEntry 3	VPN packets received
jnxJsIfMonInPolicyPermit	jnxJsIfMonEntry 4	Incoming bytes permitted by policy
jnxJsIfMonOutPolicyPermit	jnxJsIfMonEntry 5	Outgoing bytes permitted by policy



**Table 200: jnxJsIfMonTable** (continued)

Object	Object ID	Description
jnxJsIfMonConn	jnxJsIfMonEntry 6	Incoming connections established
jnxJsIfMonInMcast	jnxJsIfMonEntry 7	Multicast packets received
jnxJsIfMonOutMcast	jnxJsIfMonEntry 8	Multicast packets sent
jnxJsIfMonPolicyDeny	jnxJsIfMonEntry 9	Packets dropped due to policy denial
jnxJsIfMonNoGateParent	jnxJsIfMonEntry 10	Packets dropped due to no parent for a gate
jnxJsIfMonTcpProxyDrop	jnxJsIfMonEntry 11	Packets dropped due to syn-attack protection
jnxJsIfMonNoDip	jnxJsIfMonEntry 12	Packets dropped due to DIP errors
jnxJsIfMonNoNspTunnel	jnxJsIfMonEntry 13	Packets dropped because no NSP tunnel found
jnxJsIfMonNoNatCon	jnxJsIfMonEntry 14	Packets dropped due to no more sessions
jnxJsIfMonInvalidZone	jnxJsIfMonEntry 15	Packets dropped because an invalid zone received the packet
jnxJsIfMonIpClsFail	jnxJsIfMonEntry 16	Packets dropped due to IP classification failure
jnxJsIfMonAuthDrop	jnxJsIfMonEntry 17	Packets dropped due to user authentication errors
jnxJsIfMonMultiUserAuthDrop	jnxJsIfMonEntry 18	Packets dropped due to multiple user authentications in loopback sessions
jnxJsIfMonLoopMultiDipDrop	jnxJsIfMonEntry 19	Packets dropped due to multiple DIP in loopback sessions
jnxJsIfMonAddrSpoof	jnxJsIfMonEntry 20	Packets dropped due to address spoofing
jnxJsIfMonLpDrop	jnxJsIfMonEntry 21	Packets dropped due to no loopback
jnxJsIfMonNullZone	jnxJsIfMonEntry 22	Packets dropped due to no zone or NULL zone binding
jnxJsIfMonNoGate	jnxJsIfMonEntry 23	Packets dropped due to no NAT gateway
jnxJsIfMonNoMinorSess	jnxJsIfMonEntry 24	Packets dropped due to no minor session
jnxJsIfMonNvecErr	jnxJsIfMonEntry 25	Packets dropped due to no session for gateway
jnxJsIfMonTcpSeq	jnxJsIfMonEntry 26	Packets dropped because TCP sequence number out of window

**Table 200: jnxJsIfMonTable** (continued)

Object	Object ID	Description
jnxJsIfMonIllegalPak	jnxJsIfMonEntry 27	Packets dropped because they did not make any sense
jnxJsIfMonNoRoute	jnxJsIfMonEntry 28	Packets dropped because no route was present
jnxJsIfMonAuthFail	jnxJsIfMonEntry 29	Packets dropped because authentication failed
jnxJsIfMonSaInactive	jnxJsIfMonEntry 30	Packets dropped because security association (SA) is not active
jnxJsIfMonNoSa	jnxJsIfMonEntry 31	Packets dropped because no SA found for incoming security parameter index (SPI)
jnxJsIfMonSelfPktDrop	jnxJsIfMonEntry 32	Packets dropped because no one interested in self packets

## Chapter 71

# Interpreting the VPN Certificate Objects MIB

The enterprise-specific VPN Certificate Objects MIB, `jnxJsCertificateMIB`, whose object ID is `{jnxJsCertificates 1}`, defines the objects that are used to monitor reference and attributes to the certificates.

A related VPN Certificate Object MIB is `jnxJsCertificateObjects {jnxJsCertificateMIB 1}`.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-cert.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-cert.txt).

This chapter contains the following topics:

- `jnxJsLoadedCaCertTable` on page 613
- `jnxJsLoadedLocalCertTable` on page 614

## jnxJsLoadedCaCertTable

The `jnxJsLoadedCaCertTable`, whose object ID is `{jnxJsCertificateObjects 1}`, exposes the loaded Certification Authority (CA)-issued certificate objects listed in Table 201 on page 613. Certificates are used when establishing a secure connection in the device.

**Table 201: jnxJsLoadedCaCertTable**

Object	Object ID	Description
<code>jnxJsLoadedCaCertEntry</code>	<code>jnxJsLoadedCaCertTable 1</code>	<p>Loaded CA certificate entry. The loaded CA certificate entry is indexed by the CA certification name.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"><li>■ <code>jnxJsLoadedCaCertName</code></li><li>■ <code>jnxJsLoadedCaCertSubject</code></li><li>■ <code>jnxJsLoadedCaCertExpire</code></li><li>■ <code>jnxJsLoadedCaCertIssuer</code></li></ul>
<code>jnxJsLoadedCaCertName</code>	<code>jnxJsLoadedCaCertEntry 1</code>	Loaded CA certificate name. This name is unique within the loaded CA certificates.

**Table 201: jnxJsLoadedCaCertTable** (continued)

Object	Object ID	Description
jnxJsLoadedCaCertSubject	jnxJsLoadedCaCertEntry 2	CA certificate subject
jnxJsLoadedCaCertExpire	jnxJsLoadedCaCertEntry 3	Expiration time and date of CA certificate
jnxJsLoadedCaCertIssuer	jnxJsLoadedCaCertEntry 4	Issuer of CA certificate

## jnxJsLoadedLocalCertTable

The jnxJsLoadedLocalCertTable, whose object ID is jnxJsCertificateObjects 2, exposes the loaded local certificate objects listed in Table 202 on page 614. Certificates are used when establishing a secure connection in the device.

**Table 202: jnxJsLoadedLocalCertTable**

Object	Object ID	Description
jnxJsLoadedLocalCertEntry	jnxJsLoadedLocalCertTable 1	Default certificate entry. This entry is indexed by the certification name.  Sequence of parameters: <ul style="list-style-type: none"> <li>■ jnxJsLoadedLocalCertName</li> <li>■ jnxJsLoadedLocalCertSubject</li> <li>■ jnxJsLoadedLocalCertExpire</li> <li>■ jnxJsLoadedLocalCertIssuer</li> </ul>
jnxJsLoadedLocalCertName	jnxJsLoadedLocalCertEntry 1	Name of the local certificate. The certificate name is unique within the loaded local certificates.
jnxJsLoadedLocalCertSubject	jnxJsLoadedLocalCertEntry 2	Certificate subject.
jnxJsLoadedLocalCertExpire	jnxJsLoadedLocalCertEntry 3	Expiration time and date of the local certificate.
jnxJsLoadedLocalCertIssuer	jnxJsLoadedLocalCertEntry 4	Issuer of the local certificate.

## Chapter 72

# Interpreting the Enterprise-Specific Security Screening Objects MIB

The enterprise-specific Security Screening Objects MIB, `jnxJsScreenMIB`, whose object ID is `{jnxJsScreening 1}`, defines the MIB for the Juniper Networks Enterprise Firewall screen functionality. Juniper Networks documentation is recommended as the reference.

The Juniper Networks Security Firewall provides various detection methods and defense mechanisms to combat exploits at all stages of the path of execution, including:

- Screen option setting
- Firewall Denial-of-Service (DoS) attack
- Network DoS attack
- OS-specific DoS attack
- Fragment reassembly

Related Security Screening Objects MIBs include:

- `jnxJsScreenNotifications` `{jnxJsScreenMIB 0}`
- `jnxJsScreenObjects` `{jnxJsScreenMIB 1}`
- `jnxJsScreenTrapVars` `{jnxJsScreenMIB 2}`

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-screening.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-js-screening.txt).

This chapter contains the following topics:

- `jnxJsScreenMonTable` on page 615

## **jnxJsScreenMonTable**

---

The `jnxJsScreenMonTable`, whose object ID is `{jnxJsScreenObjects 1}`, collects the screen attributes that monitor the various attacks to enable the Juniper Networks Security Firewall to provide deep inspection (DI) protection on each of the security device's physical interfaces. These attributes are listed in Table 201 on page 613.

The screen options can be enabled at a security zone bounded to an interface or interfaces. When these options apply to traffic reaching the security device through interfaces (via a zone), they offer protection against a malicious information gathering probe or an attack to compromise, disable, or harm a network or network resources.

**Table 203: jnxJsScreenMonTable**

Object	Object ID	Description
jnxJsScreenMonEntry	jnxJsScreenMonTable 1	<p>The screen option monitoring statistics entry. Each entry is uniquely identified by the zone name.</p> <p>The data is collected on a per zone basis. There can be multiple interfaces bound to a particular zone. Hence, the statistics are aggregated across the interfaces on a per zone basis.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> <li>■ jnxJsScreenZoneName</li> <li>■ jnxJsScreenNumOff</li> <li>■ jnxJsScreenMonSynAttk</li> <li>■ jnxJsScreenMonTearDrop</li> <li>■ jnxJsScreenMonSrcRoute</li> <li>■ jnxJsScreenMonPingDeath</li> <li>■ jnxJsScreenMonAddrSpoof</li> <li>■ jnxJsScreenMonLand</li> <li>■ jnxJsScreenMonIcmpFlood</li> <li>■ jnxJsScreenMonUdpFlood</li> <li>■ jnxJsScreenMonWinnuke</li> <li>■ jnxJsScreenMonPortScan</li> <li>■ jnxJsScreenMonIpSweep</li> <li>■ jnxJsScreenMonSynFrag</li> <li>■ jnxJsScreenMonTcpNoFlag</li> <li>■ jnxJsScreenMonIpUnknownProt</li> <li>■ jnxJsScreenMonIpOptBad</li> <li>■ jnxJsScreenMonIpOptRecRt—Record route option</li> <li>■ jnxJsScreenMonIpOptTimestamp—Timestamp option</li> <li>■ jnxJsScreenMonIpOptSecurity</li> <li>■ jnxJsScreenMonIpOptLSR—Loose source route</li> <li>■ jnxJsScreenMonIpOptSSR—Strict source route</li> <li>■ jnxJsScreenMonIpOptStream—Stream options</li> <li>■ jnxJsScreenMonIcmpFrag</li> <li>■ jnxJsScreenMonIcmpLarge</li> <li>■ jnxJsScreenMonTcpSynFin</li> <li>■ jnxJsScreenMonTcpFinNoAck</li> <li>■ jnxJsScreenMonLimitSessSrc—Session limit (source IP-based)</li> </ul>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
		<ul style="list-style-type: none"> <li>■ jnxJsScreenMonLimitSessDest—Session limit (destination IP-based)</li> <li>■ jnxJsScreenMonSynAckAck</li> <li>■ jnxJsScreenMonIpFrag</li> <li>■ jnxJsScreenSynAttackThresh—Threshold data</li> <li>■ jnxJsScreenSynAttackThresh—Threshold data</li> <li>■ jnxJsScreenSynAttackTimeout—Threshold data</li> <li>■ jnxJsScreenSynAttackAlmTh—Threshold data</li> <li>■ jnxJsScreenSynAttackQueueSize—Threshold data</li> <li>■ jnxJsScreenSynAttackAgeTime—Threshold data (obsolete in this release)</li> <li>■ jnxJsScreenIcmpFloodThresh—Threshold data</li> <li>■ jnxJsScreenUdpFloodThresh—Threshold data</li> <li>■ jnxJsScreenPortScanThresh—Threshold data</li> <li>■ jnxJsScreenIpSweepThresh—Threshold data</li> <li>■ jnxJsScreenSynAckAckThres—Threshold data</li> </ul>
jnxJsScreenZoneName	jnxJsScreenMonEntry 1	Name of the security zone under which the statistics are collected
jnxJsScreenNumOfIf	jnxJsScreenMonEntry 2	Number of interfaces bound to this zone. Each counter contains the aggregated data of all the interfaces.
jnxJsScreenMonSynAttk	jnxJsScreenMonEntry 3	<p>Number of SYN (TCP connection request) attacks.</p> <p>A SYN attack is a common denial of service (DoS) technique characterized by the following pattern:</p> <ul style="list-style-type: none"> <li>■ Using a spoofed IP address not in use on the Internet, an attacker sends multiple SYN packets to the target machine.</li> <li>■ For each SYN packet received, the target machine allocates resources and sends an acknowledgement (SYN-ACK) to the source IP address. This can cause the target machine to allocate resources for more than 3 minutes to respond to just one SYN attack, subsequently wasting resources.</li> </ul>



**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonTearDrop	jnxJsScreenMonEntry 4	<p>Number of teardrop attacks.</p> <p>Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position of the data contained in a fragmented packet relative to the data of the original unfragmented packet. When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap. The server attempting to reassemble the packet can crash, especially if it is running an older operating system that has this vulnerability.</p> <p>When this option is enabled, the security device detects this discrepancy in a fragmented packet and drops it, and counts the number of packet dropped.</p>
jnxJsScreenMonSrcRoute	jnxJsScreenMonEntry 5	<p>Number of either loose source route option packets or strict source route attack packets.</p> <p>IP source route options can be used to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.</p>
jnxJsScreenMonPingDeath	jnxJsScreenMonEntry 6	<p>Number of ping-of-death attack packets.</p> <p>The maximum allowable IP packet size is 65,535 bytes, including the packet header (typically 20 bytes long). An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes.</p> <p>Many ping implementations, however, allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions, such as DoS, crashing, freezing, and rebooting.</p> <p>When the ping-of-death option is enabled, the security device detects and rejects such oversized and irregular packet sizes, even when the attacker hides the total packet size by purposefully fragmenting it.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonAddrSpoof	jnxJsScreenMonEntry 7	<p>Number of address spoofing attack packets.</p> <p>One method to gain access to a restricted network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. The mechanism to detect IP spoofing relies on route table entries.</p> <p>For example, if a packet with source IP address 10.1.1.6 arrives at port <b>eth3</b>, but the security device has a route to 10.1.1.0/24 through port <b>eth1</b>, IP spoofing checking notes that this address arrived at an invalid interface as defined in the route table. A valid packet from 10.1.1.6 can arrive only via <b>eth1</b>, not <b>eth3</b>. The security device concludes that the packet has a spoofed source IP address and discards it.</p>
jnxJsScreenMonLand	jnxJsScreenMonEntry 8	<p>Number of land attack packets.</p> <p>A SYN attack combined with an IP spoof is referred to as land attack. A land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address. The receiving victim responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the victim, causing a DoS.</p>
jnxJsScreenMonIcmpFlood	jnxJsScreenMonEntry 9	<p>Number of ICMP flood attack packets.</p> <p>An ICMP flood typically occurs when ICMP echo requests overload a victim with so many requests that the victim expends all its resources responding to the ICMP echo requests until it can no longer process valid network traffic. With ICMP flood protection enabled and a threshold set, if the threshold is exceeded, the victim invokes the flood attack protection feature.</p> <p>The default threshold value is 1000 packets per second. If the threshold is exceeded, the security device ignores further ICMP echo requests for the remainder of that second plus the next second as well.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonUdpFlood	jnxJsScreenMonEntry 10	<p>Number of UDP flood attack packets.</p> <p>UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. With UDP flood protection enabled, a threshold can be set so that when the threshold is exceeded, the system invokes UDP flood attack protection.</p> <p>The default threshold value is 1000 packets per second. If the number of UDP datagrams from one or more sources to a single destination exceeds this threshold, the security device ignores further UDP datagrams to that destination for the remainder of that second plus the next second as well.</p>
jnxJsScreenMonWinnuke	jnxJsScreenMonEntry 11	<p>Number of NetBIOS attacks.</p> <p>WinNuke is a DoS attack targeting any computer on the Internet running Microsoft Windows. The attacker sends a TCP segment, usually to NetBIOS port 139 of a host with an established connection with segment's urgent (URG) flag set. This practice introduces a NetBIOS fragment overlap, which causes many machines running Microsoft Windows to crash.</p>
jnxJsScreenMonPortScan	jnxJsScreenMonEntry 12	<p>Number of port scan attempt attack packets.</p> <p>A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval. The purpose of this attack is to scan the available services in the hope that at least one port will respond, thus identifying a service of the target. The security device should internally log the number of different ports scanned from one remote source.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonIpSweep	jnxJsScreenMonEntry 13	<p>Number of address sweep attempt attack packets.</p> <p>An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined interval. The purpose of this attack is to send ICMP packets, typically echo requests, to various hosts in the hope that at least one replies, thus uncovering an address of the target. The security device internally logs the number of ICMP packets to different addresses from one remote source.</p>
jnxJsScreenMonSynFrag	jnxJsScreenMonEntry 14	<p>Number of SYN fragments.</p> <p>IP encapsulates a TCP SYN segment in the IP packet that initiates a TCP connection. The purpose is to initiate a connection and to invoke a SYN/ACK segment response. The SYN segment typically does not contain any data since the IP packet is small and there is no legitimate reason for it to be fragmented. A fragmented SYN packet is anomalous and is suspicious. To be cautious, it might be helpful to block such fragments from entering the protected network.</p> <p>When the SYN fragmentation check is enabled, the security device detects and drops the packets when the IP header indicates that the packet has been fragmented while the SYN flag is set in the TCP header.</p>
jnxJsScreenMonTcpNoFlag	jnxJsScreenMonEntry 15	<p>Number of TCP packets with no flag set.</p> <p>A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Operating systems respond to such anomalies in different ways. The response, or even lack of response, from the targeted device can provide a clue as to the target's OS type.</p> <p>When this option is enabled, if the security device discovers such a header with a missing or malformed flags field, it drops the packet.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonIpUnknownProt	jnxJsScreenMonEntry 16	<p>Number of of unknown protocol IP packets.</p> <p>According to RFC-1700, some protocol types in an IP header are reserved and unassigned at this time. Precisely because these protocols are undefined, there is no way to know in advance whether a particular unknown protocol is benign or malicious. Unless your network makes use of a nonstandard protocol with a reserved or unassigned protocol number, a cautious stance is to block such unknown elements from entering your protected network.</p> <p>When the Unknown Protocol Protection SCREEN option is enabled, the security device drops packets when the protocol field contains a protocol ID number of 137 or greater.</p>
jnxJsScreenMonIpOptBad	jnxJsScreenMonEntry 17	<p>Number of IP bad option packets.</p> <p>The IP protocol specifies a set of eight options that provide special routing controls, diagnostic tools, and security. These eight options can be used for malicious objectives.</p> <p>Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. The incorrect formatting is anomalous and potentially harmful to the intended recipient.</p> <p>When the Bad IP Option Protection SCREEN option is enabled, the security device detects and blocks packets when any IP option in the IP packet header is incorrectly formatted.</p>
jnxJsScreenMonIpOptRecRt	jnxJsScreenMonEntry 18	<p>Number of IP record option packets.</p> <p>The IP standard RFC-791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. The record option is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a record IP option is received, the security device flags it as an network reconnaissance attack and records the event for the ingress interface.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonIpOptTimestamp	jnxJsScreenMonEntry 19	<p>Number of IP timestamp option packets.</p> <p>The IP standard RFC-791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Timestamp is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a timestamp IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIpOptSecurity	jnxJsScreenMonEntry 20	<p>Number of IP security option packets.</p> <p>The IP standard RFC-791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Security is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a security IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIpOptLSR	jnxJsScreenMonEntry 21	<p>Number of strict source route packets.</p> <p>Attackers can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonIpOptStream	jnxJsScreenMonEntry 23	<p>Number of IP stream option packets.</p> <p>The IP standard RFC-791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Stream is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a security IP option is received, the security device flags it as an network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIcmpFrag	jnxJsScreenMonEntry 24	<p>Number of ICMP fragment packets.</p> <p>ICMP provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is wrong. With the ICMP Fragment Protection SCREEN option enabled, the security device should be able to block any ICMP packet with the More Fragments flag set or with an offset value indicated in the offset field.</p>
jnxJsScreenMonIcmpLarge	jnxJsScreenMonEntry 25	<p>Number of large ICMP packets.</p> <p>Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented.</p> <p>If an ICMP packet is unusually large, something is wrong. For example, the Loki program uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a Loki agent. It might also indicate some other kind of malicious activity.</p> <p>When the the Large Size ICMP Packet Protection SCREEN option is enabled, the security device drops ICMP packets with a length greater than 1024 bytes.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonTcpSynFin	jnxJsScreenMonEntry 26	<p>Number of dropped TCP packets because SYN and FIN are both set.</p> <p>Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS.</p> <p>When the blocking of TCP packets with both SYN and FIN is enabled, the security device drops the packet when it discovers such a header.</p>
jnxJsScreenMonTcpFinNoAck	jnxJsScreenMonEntry 27	<p>Number of TCP packets with FIN set, but without the ACK bit set.</p> <p>A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response and thereby discover an active host or an active port on a host. The use of TCP segments with the FIN flag set might evade detection and thereby help attackers succeed in their reconnaissance efforts.</p>
jnxJsScreenMonLimitSessSrc	jnxJsScreenMonEntry 28	<p>Number of the session connections for a source IP address that exceeds the specified limit.</p> <p>Because all the virus-generated traffic originates from the same IP address (generally from an infected server), a source-based session limit ensures that the firewall can curb such excessive amounts of traffic. This amount is based on a threshold value of the number of concurrent sessions required to fill up the session table of the particular firewall.</p> <p>The default maximum for a source-based session limit is 128 concurrent sessions, which can be adjusted accordingly.</p>



**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonLimitSessDest	jnxJsScreenMonEntry 29	<p>Number of session connections for the destination source IP address that exceeds the specified limit.</p> <p>The user can limit the number of concurrent sessions to the same destination IP address. An attacker can launch a distributed denial-of-service (DDoS) attack using “zombie agents.” Setting a destination-based session limit can ensure that the security device allows only an acceptable number of concurrent connection requests, no matter what the source, to reach any one host.</p> <p>The default maximum for the destination-based session limit is 128 concurrent sessions.</p>
jnxJsScreenMonSynAckAck	jnxJsScreenMonEntry 30	<p>Number of SYN ACK ACK attacks.</p> <p>When an authentication user initiates a Telnet or FTP connection, the user sends a SYN segment to the Telnet or FTP server. The security device intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At that point, the initial three-way handshake is complete. The security device sends a login prompt to the user. When a malicious user does not log in, but instead continues initiating SYN-ACK-ACK sessions, the firewall session table can fill up to the point at which the security device begins rejecting legitimate connection requests.</p> <p>When the SYN-ACK-ACK proxy protection option is enabled, after the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, the security device rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenMonIpFrag	jnxJsScreenMonEntry 31	<p>Number of block IP fragment packets.</p> <p>As a packets travels, it is sometimes necessary to break the packet into smaller fragments based upon the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system.</p> <p>When the block IP fragmentation flag is enabled, the security device blocks all IP packet fragments that it receives at interfaces bound to that zone.</p>
<b>Threshold Values</b>		
jnxJsScreenSynAttackThresh	jnxJsScreenMonEntry 32	<p>SYN attack threshold value.</p> <p>The number of SYN segments to the same destination address and port number per second required to activate the SYN proxying mechanism. In order to set the appropriate threshold value, it requires a through knowledge of the normal traffic patterns at the site.</p> <p>For example, if the security device normally gets 2000 SYN segments per second, the threshold value should be set at 3000 segments per second.</p>
jnxJsScreenSynAttackTimeout	jnxJsScreenMonEntry 33	<p>SYN attack timeout value.</p> <p>The maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds.</p>
jnxJsScreenSynAttackAlmTh	jnxJsScreenMonEntry 34	<p>SYN attack alarm threshold value.</p> <p>The SYN attack alarm threshold causes an alarm to be generated when the number of proxied, half-completed TCP connection requests per second to the same destination address and port number exceeds its value.</p>
jnxJsScreenSynAttackQueueSize	jnxJsScreenMonEntry 35	<p>SYN attack queue size.</p> <p>The number of proxied connection requests held in the proxied connection queue before the security device starts rejecting new connection requests.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
<b>NOTE:</b> The jnxJsScreenSynAttackAgeTime object is obsolete in this release.		
jnxJsScreenSynAttackAgeTime	jnxJsScreenMonEntry 36	SYN flood age time
jnxJsScreenIcmpFloodThresh	jnxJsScreenMonEntry 37	<p>ICMP attack alarm threshold value.</p> <p>The security device can impose a limit on the number of SYN segments permitted to pass through the firewall per second. The default attack threshold value is 1000. The valid threshold range is 1 through 100000. When the threshold value is exceed, an alarm is triggered.</p>
jnxJsScreenUdpFloodThresh	jnxJsScreenMonEntry 38	<p>UDP attack alarm threshold value.</p> <p>UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections.</p> <p>The default threshold value is 1000 packets per second.</p>
jnxJsScreenPortScanThresh	jnxJsScreenMonEntry 39	<p>Port scan threshold value.</p> <p>The port scan threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1000000.</p> <p>By using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), the security device flags this occurrence as a port scan attack and rejects all further packets from the remote source for the remainder of the specified timeout period. The security device detects and drops the tenth packet that meets the port scan attack criterion.</p>

**Table 203: jnxJsScreenMonTable** (continued)

Object	Object ID	Description
jnxJsScreenIpSweepThresh	jnxJsScreenMonEntry 40	<p>IP sweep threshold interval.</p> <p>The IP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1000000.</p> <p>By using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this occurrence as an address sweep attack and rejects all further ICMP echo requests from that host for the remainder of the specified threshold time period. The security device detects and drops the tenth packet that meets the address sweep attack criterion.</p>
jnxJsScreenSynAckAckThres	jnxJsScreenMonEntry 41	SYN-ACK-ACK alarm threshold value

## Chapter 73

# Interpreting the Enterprise-Specific LDP MIB

The enterprise-specific Label Distribution Path (LDP) MIB, whose object identifier is {jnxMibs 14}, contains LDP statistics, and defines LDP notification objects and types.

The enterprise-specific LDP MIB uses the following objects and definitions from standard MIBs and enterprise-specific MIB definitions:

- `IpAddress` from SNMPv2-SMI MIB
- `DisplayString` from SNMPv2-TC MIB
- `InterfaceIndex` and `InterfaceIndexOrzero` from IF MIB
- `jnxMibs` and `jnxLdpTraps` from Juniper Enterprise-Specific SMI MIB
- `jnxMplsLdpSesState` from Juniper Enterprise-Specific MPLS LDP MIB
- `MplsVpnName` from the standard MPLS VPN MIB
- `InetAddressType`, `InetAddress`, and `InetAddressPrefixLength` from the standard Inet Address MIB

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ldp.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ldp.txt).

This chapter contains the following sections:

- LDP Notification Objects and Notification Types on page 631
- LDP Statistics Table on page 634

## LDP Notification Objects and Notification Types

---

The `jnxLdpTrapVars` table, whose object identifier is {jnxLdp 1}, defines the objects used in the enterprise-specific LDP traps.



**NOTE:** The enterprise-specific LDP MIB requires `jnxLdpTrapPrefix` with 0 subidentifier for seamless translation of SNMPv2 notifications to SNMPv1 format.

---

**Table 204: LDP Notification Objects**

Object	Object ID	Description
jnxLdpLspFec	jnxLdpTrapVars 1	Contains the LSP forwarding equivalence class (FEC) in <b>IpAddress</b> format.
jnxLdpRtrid	jnxLdpTrapVars 2	Contains the router ID of the sending router in <b>IpAddress</b> format.
jnxLdpLspDownReason	jnxLdpTrapVars 3	<p>Returns one of the following integer values to denote what might have caused the LSP to go down:</p> <ul style="list-style-type: none"> <li>■ 1–Change in topology</li> <li>■ 2–Withdrawal of label by the neighbor</li> <li>■ 3–Unavailability of the neighbor because the neighbor is down</li> <li>■ 4–Change in filter</li> <li>■ 5–Unknown reason</li> </ul>
jnxLdpSesDownReason	jnxLdpTrapVars 4	<p>Returns one of the following integer values to denote what might have caused the session transition into non-existent state:</p> <ul style="list-style-type: none"> <li>■ 0–Unknown reason</li> <li>■ 1–Hold time expired</li> <li>■ 2–Connection time expired</li> <li>■ 3–All adjacencies are down</li> <li>■ 4–Received bad TLV (type, length, and value encoding scheme)</li> <li>■ 5–Received bad PDU (protocol data unit)</li> <li>■ 6–Connection error occurred</li> <li>■ 7–The connection was reset</li> <li>■ 8–Received notification from peer</li> <li>■ 9–Received unexpected end-of-file message</li> <li>■ 10–The authentication key was changed</li> <li>■ 11–Error occurred during initialization</li> <li>■ 12–Graceful restart was aborted</li> <li>■ 13– CLI command was issued to end the session</li> </ul>
jnxLdpSesDownIf	jnxLdpTrapVars 5	Contains the SNMP index of the interface associated with the session-down event. If no interface is associated with the session-down event, then this object returns the SNMP index of any interface associated with one of the neighbors.
jnxLdpLspFecLen	jnxLdpTrapVars 6	Represents the length of the LSP FEC prefix in bits. The allowable range is 0 through 32 bits.
jnxLdpSesUpIf	jnxLdpTrapVars 7	Contains the interface index of one of the neighbors associated with the session.
jnxLdpInstanceName	jnxLdpTrapVars 8	Contains the name of the VPN interface.

Table 205 on page 633 lists the enterprise-specific LDP notifications and the objects contained in each notification. The enterprise-specific LDP notifications use the objects listed in Table 204 on page 632.

**Table 205: LDP Notification Types**

Object	Object ID	Description
jnxLdpLspUp	jnxLdpTrapPrefix 1	<p>Generated when an LSP comes back online. Typically, this trap is generated only when an LSP that has an active jnxLdpLspDown trap comes back online.</p> <ul style="list-style-type: none"> <li>■ jnxLdpLspFec</li> <li>■ jnxLdpRtrid</li> <li>■ jnxLdpLspFecLen</li> <li>■ jnxLdpInstanceName</li> </ul>
jnxLdpLspDown	jnxLdpTrapPrefix 2	<p>Generated when an LSP goes offline. This trap contains the following objects:</p> <ul style="list-style-type: none"> <li>■ jnxLdpLspFec</li> <li>■ jnxLdpRtrid</li> <li>■ jnxLdpLspDownReason</li> <li>■ jnxLdpLspFecLen</li> <li>■ jnxLdpInstanceName</li> </ul> <p><b>NOTE:</b> For every jnxLdpLspDown trap generated, JUNOS software ensures that a jnxLdpLspUp trap is generated when the LSP comes back online.</p>
jnxLdpSesUp	jnxLdpTrapPrefix 3	<p>Generated when the jnxMplsLdpSesState object moves into the <b>operational</b> (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> <li>■ jnxMplsLdpSesState</li> <li>■ jnxLdpSesUpIf</li> </ul>
jnxLdpSesDown	jnxLdpTrapPrefix 4	<p>Generated when the jnxMplsLdpSesState object moves out of the <b>operational</b> (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> <li>■ jnxMplsLdpSesState</li> <li>■ jnxLdpSesDownReason</li> <li>■ jnxLdpSesDownIf</li> </ul> <p>The jnxLdpSesDownIf object contains the address of the interface associated with the last neighbor when the value of jnxLdpSesDownReason was allAdjacenciesDown (3).</p>

## LDP Statistics Table

The `jnxLdpStatsTable`, whose object identifier is `{jnxLdp 2}`, contains the statistics associated with a particular LDP FEC. Each `jnxLdpStatsEntry` in `jnxLdpStatsTable` contains the objects listed in Table 206 on page 634.

**Table 206: jnxLdpStatsTable**

Object	Object ID	Description
<code>jnxLdpInstanceId</code>	<code>jnxLdpStatsEntry 1</code>	Identifies the LDP instance
<code>jnxLdpFecType</code>	<code>jnxLdpStatsEntry 2</code>	Denotes the type of the LDP instance.
<code>jnxLdpFec</code>	<code>jnxLdpStatsEntry 3</code>	Contains the <code>InetAddress</code> of the LDP FEC.
<code>jnxLdpFecLength</code>	<code>jnxLdpStatsEntry 4</code>	Shows the LDP FEC length in bits. The allowable range is 0 through 32 bits.
<code>jnxLdpFecStatisticsStatus</code>	<code>jnxLdpStatsEntry 5</code>	<p>Contains one of the following integer values to indicate the status of traffic statistics for the FEC:</p> <ul style="list-style-type: none"> <li>■ 1—Enabled and available</li> <li>■ 2—Disabled</li> <li>■ 3—Unavailable</li> </ul> <p>The traffic statistics may be disabled for the penultimate hop FECs, and in such cases, the objects after <code>jnxLdpFecStatisticsStatus</code> in the <code>jnxLdpStatsEntry</code> return 0 value.</p>
<code>jnxLdpIngressOctets</code>	<code>jnxLdpStatsEntry 6</code>	Shows the number of octets of traffic originated from the router, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object may return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
<code>jnxLdpIngressPackets</code>	<code>jnxLdpStatsEntry 7</code>	Shows the number of packets originated from the router, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object may return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
<code>jnxLdpTransitOctets</code>	<code>jnxLdpStatsEntry 8</code>	Shows the number of octets of traffic originated from a different router but destined for this FEC, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object may return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
<code>jnxLdpTransitPackets</code>	<code>jnxLdpStatsEntry 9</code>	Shows the number of packets of traffic originated from a different router but destined for this FEC, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object may return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.



## Chapter 74

# Interpreting the Enterprise-Specific EX-Series SMI MIB

The enterprise-specific Structure of Management Information (SMI) management information base (MIB) for EX-series leverages the `jnxExMibRoot` object from the enterprise-specific SMI MIB (`jnx-smi.mib`), and defines a MIB branch for switching-related MIB definitions for the EX-series Ethernet switches. MIB objects that are specific to EX-series are identified with a `jnxEx` prefix.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ex-smi.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-ex-smi.txt).

The `jnxExMibRoot` contains one branch, `jnxExSwitching`, whose object identifier is `{jnxExMibRoot 1}`.

The `jnxExSwitching` branch contains the objects listed in Table 207 on page 635.

**Table 207: jnxExSwitching**

Object	Object ID	Description
<code>jnxExAnalyzer</code>	<code>jnxExSwitching 1</code>	Defines the branch for the enterprise-specific Analyzer MIB. For more information on Analyzer MIB, see “Interpreting the Enterprise-Specific Analyzer MIB” on page 637.
<code>jnxExSecureAccessPort</code>	<code>jnxExSwitching 2</code>	Defines the branch for the enterprise-specific Secure Access Port MIB. For more information on Secure Access Port MIB, see “Interpreting the Enterprise-Specific Secure Access Port MIB” on page 651.
<code>jnxExPaeExtention</code>	<code>jnxExSwitching 3</code>	Defines the branch for the enterprise-specific PAE Extensions MIB. For more information on PAE Extensions MIB, see “Interpreting the Enterprise-Specific PAE Extension MIB” on page 647.
<code>jnxExVirtualChassis</code>	<code>jnxExSwitching 4</code>	Defines the branch for the enterprise-specific Virtual Chassis MIB. For more information on Virtual Chassis MIB, see “Interpreting the Enterprise-Specific Virtual Chassis MIB” on page 645.
<code>jnxExVlan</code>	<code>jnxExSwitching 5</code>	Defines the branch for the enterprise-specific VLAN MIB. For more information on VLAN MIB, see “Interpreting the Enterprise-Specific VLAN MIB” on page 641.



## Chapter 75

# Interpreting the Enterprise-Specific Analyzer MIB

The Juniper Networks enterprise-specific Analyzer MIB, whose object identifier is {jnxExAnalyzer 1}, contains analyzer and remote analyzer data related to port mirroring on the EX-series Ethernet switches. Port mirroring is a method used on enterprise switches to monitor and analyze traffic on the network.

When port mirroring is enabled, copies of all (or a sample set of) packets are forwarded from one port of the switch to another port on the same switch (analyzer) or on another switch (remote analyzer) where the packet can be analyzed and studied.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-analyzer.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-analyzer.txt).

This chapter contains the following sections:

- Analyzer Table on page 637
- Analyzer Input Table on page 638
- Analyzer Output Table on page 639

## Analyzer Table

The jnxAnalyzerTable, whose object identifier is {jnxAnalyzerMIBObjects 1}, contains information on analyzer parameters. Each jnxAnalyzerEntry contains the objects listed in Table 208 on page 637.

**Table 208: jnxAnalyzerTable**

Object	Object ID	Description
jnxAnalyzerName	jnxAnalyzerEntry 1	Uniquely identifies an analyzer configured on the switch.
jnxAnalyzerStatus	jnxAnalyzerEntry 2	Shows whether mirroring is enabled or disabled on the analyzer.
jnxMirroringRatio	jnxAnalyzerEntry 3	Specifies the mirroring ratio. This object defines the sample size for mirroring. For example, 1 out of every x packets, where x is a number in the range of 1 through 2047.

**Table 208: jnxAnalyzerTable (continued)**

Object	Object ID	Description
jnxLossPriority	jnxAnalyzerEntry 4	<p>Specifies the loss priority for a packet. When the number of packets at the analyzer port exceeds the bandwidth of the analyzer port, packets are dropped based on the loss priority value. When there is a bandwidth crunch, packets with high loss priority are dropped to clear the congestion. This object uses the following integer values to denote the loss priority:</p> <ul style="list-style-type: none"> <li>■ 0—Low loss priority</li> <li>■ 1—High loss priority</li> </ul>

## Analyzer Input Table

The `jnxAnalyzerInputTable`, whose object identifier is `{jnxAnalyzerMIBObjects 2}`, contains information about analyzer sessions. In a typical analyzer session, several source ports can be associated with a single destination port, and a range or series of ports can be mirrored.

Each `jnxAnalyzerInputEntry` provides information about input source ports, and contains the objects listed in Table 209 on page 638.

**Table 209: jnxAnalyzerInputTable**

Object	Object ID	Description
jnxAnalyzerInputValue	jnxAnalyzerInputEntry 1	<p>Identifies an analyzer input source port. This object can contain a display string of not more than 255 characters.</p> <ul style="list-style-type: none"> <li>■ If the value of <code>jnxAnalyzerInputType</code> is 1, then the value of <code>jnxAnalyzerInputValue</code> denotes the interface name of the input source.</li> <li>■ If the value of <code>jnxAnalyzerInputType</code> is 2, then the value of <code>jnxAnalyzerInputValue</code> denotes the VLAN name of the input source.</li> </ul>
jnxAnalyzerInputOption	jnxAnalyzerInputEntry 2	<p>Denotes the type of traffic to be mirrored from the source port; that is, whether it is ingress traffic or egress traffic. This object uses the following integer values:</p> <ul style="list-style-type: none"> <li>■ 1—Ingress traffic, where the analyzer monitors packets received by the source port.</li> <li>■ 2—Egress traffic, where the analyzer monitors packets transmitted by the source port.</li> </ul> <p>In both the cases, the number of packets mirrored to the destination port depends on the <code>jnxMirroringRatio</code>.</p>
jnxAnalyzerInputType	jnxAnalyzerInputEntry 3	<p>Denotes whether the mirroring source is an interface or a VLAN. This object uses integer values 1 (for interface) and 2 (for VLAN).</p> <p>For interfaces, you can configure either ingress or egress mirroring, whereas, for VLANs, you can configure only ingress mirroring.</p>

Analyzer Output Table

The jnxAnalyzerOutputTable, whose object identifier is {jnxAnalyzerMIBObjects 3}, contains information about destination port to which the packets are mirrored. Each jnxAnalyzerOutputEntry contains the objects listed in Table 210 on page 639, and provides information about destination port or destination VLAN.

Table 210: jnxAnalyzerOutputTable

Object	Object ID	Description
jnxAnalyzerOutputValue	jnxAnalyzerOutputEntry 1	Uniquely identifies a destination port or VLAN. This object can contain a string of not more than 255 characters.  If the value of jnxAnalyzerOutputType is 1, then jnxAnalyzerOutputValue contains an interface name.  If the value of jnxAnalyzerOutputType is 1, then jnxAnalyzerOutputValue contains a VLAN name.
jnxAnalyzerOutputType	jnxAnalyzerOutputEntry 2	Denotes the type of the output destination port. This object uses integer values 1 (for destination port that is on the same switch) and 2 (for remote analyzer, that is a dedicated VLAN on a different switch).



## Chapter 76

# Interpreting the Enterprise-Specific VLAN MIB

The enterprise-specific VLAN MIB for EX-series Ethernet switches, whose object identifier is {jnxExSwitching 5}, contains information about prestandard IEEE 802.10 VLANs and their association with LAN Emulation Clients (LAC). Devices with prestandard implementation maintain port groupings and associated filters that are used to form a virtual bridge.

The enterprise-specific VLAN MIB leverages the following objects and data types from standard MIBs, RFCs, and Juniper Networks enterprise-specific MIBs:

- Integer 32 and IpAddress—From SNMPv2-SMI
- MacAddress, DisplayString, and TruthValue—From SNMPv2-TC
- InterfaceIndex—From IF MIB
- InetAddress and InetAddressType—From Inet Address MIB
- jnxExVlan—From Juniper Networks enterprise-specific SMI MIB

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vlan.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-vlan.txt)

This section contains the following topics:

- VLAN Configuration Table on page 641
- VLAN Interfaces Table on page 642
- Port Group Table on page 643
- MAC List Table on page 643

## VLAN Configuration Table

---

The jnxVlanTable, whose object identifier is {jnxVlanMibObjects 1}, contains VLAN names and properties. Each jnxVlanEntry contains the entries listed in Table 211 on page 642.

**Table 211: jnxVlanTable**

Object	Object ID	Description
jnxVlanName	jnxVlanEntry 1	Contains the name of the VLAN, VLAN name can be a string of not more than 255 characters.
jnxVlanID	jnxVlanEntry 2	Contains the identifier used internally by the device to reference the VLAN. This object can be an integer value in the range 1 through 4094.
jnxVlanType	jnxVlanEntry 3	Denotes the type of the VLAN. This object uses the following integer values: <ul style="list-style-type: none"> <li>■ 1–Static</li> <li>■ 2–Dynamic</li> </ul> <p>The default value for this object is 1, static.</p>
jnxVlanPortGroupInstance	jnxVlanEntry 4	Contains the index that identifies the subtree in
jnxVlanMacListInstance	jnxVlanEntry 5	Contains an index that identifies the subtree to retrieve the list of MAC addresses to the jnxVlanMacListTable subtree to retrieve in

## VLAN Interfaces Table

The jnxVlanInterfaceTable, whose object identifier is {jnxVlanMIBObjects 2 }, contains information about the Layer 3 properties of VLANs. Each jnxVlanInterfaceEntry, indexed with jnxVlanName, contains the objects listed in Table 212 on page 642.

**Table 212: jnxVlanInterfaceTable**

Object	Object ID	Description
jnxVlanInterfaceIpAddress	jnxVlanInterfaceEntry 1	Contains the IP address of the interface.
jnxVlanInterfaceProtocol	jnxVlanInterfaceEntry 2	Specifies the protocol used.
jnxVlanInterfaceSubNetMask	jnxVlanInterfaceEntry 3	Specifies the subnet mask of the VLAN,
jnxVlanInterfaceBroadcastAddress	jnxVlanInterfaceEntry 4	Specifies the broadband address of the VLAN.
jnxVlanInterfaceDescription	jnxVlanInterfaceEntry 5	Contains a description for the VLAN. This object can contain a string of not more than 255 characters.
jnxVlanInterfaceAdminStatus	jnxVlanInterfaceEntry 6	Denotes the administration status of the VLAN.
jnxVlanInterfaceOperStatus	jnxVlanInterfaceEntry 7	Denotes the operational status of the VLAN.
jnxVlanSnmplfIndex	jnxVlanInterfaceEntry 8	Specifies the SNMP IF Index for the interface.



## Port Group Table

The `jnxVlanPortGroupTable` contains information about port groupings. Each `jnxVlanPortGroupEntry` contains the objects listed in Table 213 on page 643.

**Table 213: jnxVlanPortGroupTable**

Object	Object ID	Description
<code>jnxVlanPortGroupIndex</code>	<code>jnxVlanPortGroupEntry 1</code>	Uniquely identifies a port group.
<code>jnxVlanPort</code>	<code>jnxVlanPortGroupEntry 2</code>	Specifies the port on the VLAN with which this port group is associated.
<code>jnxVlanPortStatus</code>	<code>jnxVlanPortGroupEntry 3</code>	Shows the status of association between the port and the VLAN. This object uses the following integer values: <ul style="list-style-type: none"> <li>■ <b>1–autoActive:</b> The port is part of the VLAN because the switch has automatically added the port.</li> <li>■ <b>2–allowed:</b> The port has been configured to be part of the VLAN, and will be allowed to be part of the VLAN, if the port meets all other requirements.</li> <li>■ <b>3–allowedActive:</b> The port has been configured to be part of the VLAN, and will be allowed to be part of the VLAN, if the port meets all other requirements. However, unlike in the case of <b>allowed</b> ports, this port has a device that is participating in the VLAN associated with the port.</li> <li>■ <b>4–allowedNotAvail:</b> The port is active on some other VLAN, and is not available currently. This value applies to devices that do not allow a port to be part of more than one VLANs at the same time.</li> <li>■ <b>5–notAssociated:</b> The port is part of a port group that is not associated with the VLAN.</li> </ul>

## MAC List Table

The `jnxVlanMacListTable`, whose object identifier is `{jnxVlanMIBObjects 4}`, contains information about MAC address lists. Each `jnxVlanMacListEntry` contains the objects listed in Table 214 on page 644.

**Table 214: jnxVlanMacListTable**

Object	Object ID	Description
jnxVlanMacListIndex	jnxVlanMacListEntry 1	Uniquely identifies a MAC address list.
jnxVlanMacAddress	jnxVlanMacListEntry 2	Specifies a MAC address that belongs to the group.

## Chapter 77

# Interpreting the Enterprise-Specific Virtual Chassis MIB

The enterprise-specific Virtual Chassis MIB, whose object identifier is {jnxExSwitching 4} contains information about virtual chassis on EX-series Ethernet switches. EX 4200 switches allow you to connect two or more switches (maximum ten) together to form a virtual chassis that can be managed as a single network element. The switches can be connected through dedicated 64 Gbps virtual chassis ports (VCPs) or through 10 Gbps fiber uplink ports.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-virtualchassis.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-virtualchassis.txt).

This chapter contains the following section:

- Virtual Chassis Member Table on page 645

## Virtual Chassis Member Table

The jnxVirtualChassisMemberTable, whose object identifier is {jnxVirtualChassisMemberMIB 1}, contains information about the devices that form the virtual chassis. Each jnxVirtualChassisMemberEntry contains the objects listed in Table 215 on page 645.

**Table 215: jnxVirtualChassisMemberTable**

Object	Object ID	Description
jnxVirtualChassisMemberId	jnxVirtualChassisMemberEntry 1	Uniquely identifies a virtual chassis member. This object contains integer values in the range 0 through 9.
jnxVirtualChassisMemberSerialNumber	jnxVirtualChassisMemberEntry 2	Contains the serial number of the virtual chassis member.
jnxVirtualChassisMemberRole	jnxVirtualChassisMemberEntry 3	Specifies the type of virtual chassis member. This object uses the following integer values: <ul style="list-style-type: none"><li>■ 1–Master</li><li>■ 2–Backup</li><li>■ 3–Linecard</li></ul>

**Table 215: jnxVirtualChassisMemberTable** (continued)

Object	Object ID	Description
jnxVirtualChassisMemberMacAddBase	jnxVirtualChassisMemberEntry 4	Specifies the media access control (MAC) address base for the virtual-chassis member.
jnxVirtualChassisMemberSWVersion	jnxVirtualChassisMemberEntry 5	Identifies the JUNOS Base operating system software suite that is installed on the virtual chassis member.
jnxVirtualChassisMemberPriority	jnxVirtualChassisMemberEntry 6	Specifies the priority of the virtual-chassis member. This object contains integer values in the range <b>1</b> through <b>255</b> .
jnxVirtualChassisMemberUptime	jnxVirtualChassisMemberEntry 7	Specifies the virtual chassis member uptime.

## Chapter 78

# Interpreting the Enterprise-Specific PAE Extension MIB

The enterprise-specific Port Access Entity (PAE) Extension MIB, whose object identifier is {jnxExSwitching 3}, is an extension of the standard IEEE802.1x PAE Extension MIB, and contains information for Static MAC Authentication. The enterprise-specific PAE Extension MIB has two branches, jnxPaeExtensionMIBNotification and jnxPaeExtensionMIBObjects.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pae-extension.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-pae-extension.txt).

This chapter contains the following sections:

- jnxAuthProfileName on page 647
- Authentication Configuration Extension Table on page 647
- Static MAC List Authentication Bypass Table on page 648
- jnxStaticMacAuthBypassIfTable on page 648

## jnxAuthProfileName

The jnxAuthProfileName object, whose object identifier is {jnxPaeExtensionMIBObjects 1}, contains the authentication profile name. The authentication profile contains the RADIUS server IP address, port number, and the secret key.

## Authentication Configuration Extension Table

The jnxPaeAuthConfigTable, whose object identifier is {jnxPaeExtensionMIBObjects 2}, contains the configuration objects for the Authenticator PAE associated with each port. Each jnxPaeAuthConfigEntry, indexed with dot1xPaePortNumber from the standard IEEE802.1x PAE Extension MIB, contains the objects listed in Table 216 on page 647.

**Table 216: jnxPaeAuthConfigTable**

Object	Object ID	Description
jnxPaeAuthConfigMacAuthStatus	jnxPaeAuthConfigEntry 1	Shows whether MAC authentication is enabled on the specified PAE port.

**Table 216: jnxPaeAuthConfigTable** (continued)

Object	Object ID	Description
jnxPaeAuthConfigGuestVlan	jnxPaeAuthConfigEntry 2	Specifies the VLAN to which an unauthenticated client moves.  This object can contain a string of not more than 255 characters.
jnxPaeAuthConfigNumberRetries	jnxPaeAuthConfigEntry 3	Specifies the maximum number of failed authentication retries allowed on an interface before the interface transitions into quiet period. No authentication happens on the interface during the quiet period.

## Static MAC List Authentication Bypass Table

The `jnxStaticMacAuthBypassTable`, whose object identifier is `{jnxPaeExtensionMIBObjects 3}`, contains a static list of MAC addresses specified by a user. The static MAC address list contains the MAC addresses of clients associated with a port. The clients whose MAC addresses are in the MAC address list are allowed to connect to the port without authentication. 802.1X or MAC authentication process is initiated for a connection request only when a matching entry is not available for the client in the `jnxStaticMacAuthBypassTable`.

The `jnxStaticMacAuthBypassTable` allows devices like printers that do not support 802.1X to connect to 802.1X-enabled ports.

Each `jnxStaticMacAuthBypassEntry`, whose object identifier is `{jnxStaticMacAuthBypassTable 1}`, contains the objects listed in Table 217 on page 648.

**Table 217: jnxStaticMacAuthBypassTable**

Object	Object ID	Description
jnxStaticMacAddress	jnxStaticMacAuthBypassEntry 1	Specifies the MAC address of the client connected to the PAE port.
jnxStaticMacVlanName	jnxStaticMacAuthBypassEntry 2	Specifies the VLAN to which the client is assigned.

## jnxStaticMacAuthBypassIfTable

The `jnxStaticMacAuthBypassIfTable`, whose object identifier is `{jnxPaeExtensionMIBObjects 4}`, contains a list of interfaces associated with the MAC addresses in the `jnxStaticMacAuthBypassTable`.

Each `jnxStaticMacAuthBypassIfEntry`, whose object identifier is `{jnxStaticMacAuthBypassIfTable 1}`, is indexed with `jnxStaticMacAddress` (from `jnxStaticMacAuthBypassTable`) and `jnxStaticMacIfIndex`. The `jnxStaticMacIfIndex` contains

a list of interfaces from which a MAC address is allowed. If the interface associated with a MAC address does not match with the one stored in this entry, the authentication bypass does not happen.





## Chapter 79

# Interpreting the Enterprise-Specific Secure Access Port MIB

The enterprise-specific Secure Access Port MIB for EX-series Ethernet switches, whose object identifier is `{jnxExSwitching }`, contains information about secure access port configuration. The EX-series Ethernet switches use DHCP snooping and dynamic ARP inspection mechanisms to extend security capabilities on the interfaces.

The enterprise-specific Secure Access Port MIB also supports features like MAC address limiting, IP Source Guard, MAC Source Guard, and Storm Control.

For a downloadable version of this MIB, see [www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-secure-access-port.txt](http://www.juniper.net/techpubs/software/junos/junos92/swconfig-net-mgmt/mib-jnx-secure-access-port.txt).

This chapter contains the following sections:

- Port Security Table for VLAN on page 651
- Port Security Table for Interface on page 652
- Storm Control Table on page 653
- DHCP Snooping Notification on page 654
- MAC Limit Exceeded Notification on page 654
- Storm Event Notification on page 654

## Port Security Table for VLAN

---

The `jnxSecAccessPortVlanTable`, whose object identifier is `{jnxSecAccessPortMIBObjects 1}`, contains information related to DHCP Snooping and dynamic ARP inspection configuration for VLANs. When a VLAN gets added to a device that supports `jnxSecAccessPortVlanTable`, a corresponding `jnxSecAccessPortVlanEntry` gets added to the table. Each `jnxSecAccessPortVlanEntry` indicates whether DHCP Snooping and dynamic ARP inspection are enabled or disabled for a VLAN, and contains the objects listed in Table 218 on page 652.

**Table 218: jnxSecAccessPortVlanTable**

Object	Object ID	Description
jnxSecAccessVlanName	jnxSecAccessPortVlanEntry 1	Contains the name of the VLAN to which the jnxSecAccessPortVlanEntry maps. This object can contain a string of not more than 255 characters.
jnxSecAccessVlanDhcpSnoopStatus	jnxSecAccessPortVlanEntry 2	Shows whether DHCP snooping is enabled ( <b>true</b> ) or disabled ( <b>false</b> ) on the VLAN
jnxSecAccessVlanDAIStatus	jnxSecAccessPortVlanEntry 3	Shows whether dynamic ARP inspection is enabled ( <b>true</b> ) or disabled ( <b>false</b> ) on the VLAN.

## Port Security Table for Interface

The jnxSecAccessPortIfTable, whose object identifier is {jnxSecAccessPortMIBObjects 2}, contains the following information:

- Trust state and rate limit of each interface for DHCP snooping
- Maximum number of MAC addresses that the interface can learn
- IP source guard and MAC source guard status for each interface

Each jnxSecAccessPortIfEntry contains the objects listed in Table 219 on page 652.

**Table 219: jnxSecAccessPortIfTable**

Object	Object ID	Description
jnxSecAccessdsIfTrustState	jnxSecAccessPortIfEntry 1	Contains one of the following values to indicate whether the interface is trusted for DHCP snooping: <ul style="list-style-type: none"> <li>■ <b>true</b>—the interface is trusted; that is, the packets coming to the interfaces are forwarded without checking.</li> <li>■ <b>false</b>—the interface is not trusted; that is, the packets coming to the interface are subjected to DHCP checks.</li> </ul>
jnxSecAccessdsIfRateLimit	jnxSecAccessPortIfEntry 2	Indicates the rate limit value for DHCP snooping. The rate limit is specified in packets per second unit. A value of 0 indicates that no rate limit is applied for DHCP traffic on the interface.
jnxSecAccessIfMacLimit	jnxSecAccessPortIfEntry 3	Specifies the maximum number of MAC address entries allowed on the interface. The default value is 5. A value of 0 indicates that no threshold limit is set for the interface. When the value for this object is 0, the value of the corresponding jnxSecAccessIfMacLimitExceed does not have any effect.

**Table 219: jnxSecAccessPortIfTable** (continued)

Object	Object ID	Description
jnxSecAccessIfMacLimitExceed	jnxSecAccessPortIfEntry 4	<p>Specifies the action to be taken when the number of MAC addresses exceeds the value set for jnxSecAccessIfMacLimit.</p> <p>This object returns one of the following JnxMacLimitExceededAction values:</p> <ul style="list-style-type: none"> <li>■ <b>1–none</b>: Indicates that no MAC address limit is set for the interface, and that no action is required.</li> <li>■ <b>2–drop</b>: Disables the MAC address learning on the interface because the number of MAC addresses has exceeded the maximum limit. Generates a notification to indicate that the number of MAC addresses has exceeded the maximum number. MAC address learning restarts only after the number of MAC addresses returns to a value within the maximum allowed number.</li> <li>■ <b>3–alarm</b>: Generates a notification to indicate that the number of MAC addresses has exceeded the maximum limit.</li> <li>■ <b>4–shutdown</b>: Blocks the traffic on the interface because the number of MAC addresses has exceeded the maximum limit. Generates a notification to indicate the status.</li> </ul> <p><b>NOTE:</b> The value for this object is invalid if jnxIfMacLimit is set to 0.</p>
jnxSecAccessIfIpSrcGuardStatus	jnxSecAccessPortIfEntry 5	Indicates whether IP source guard is enabled ( <b>true</b> ) or disabled ( <b>false</b> ) on the interface
jnxSecAccessIfMacSrcGuardStatus	jnxSecAccessPortIfEntry 6	Indicates whether MAC source guard is enabled ( <b>true</b> ) or disabled ( <b>false</b> ) on the interface.

## Storm Control Table

The jnxStormCtlTable, whose object identifier is {jnxSecAccessPortMIBObjects 3} contains information about different traffic types, the storm control parameters such as rising threshold and falling threshold associated with each type, and the action to be taken when the traffic exceeds the rising threshold level.

Each jnxStormCtlEntry, indexed with ifIndex and jnxStormCtlIfTrafficType, contains the objects listed in Table 220 on page 653.

**Table 220: jnxStormCtlTable**

Object	Object ID	Description
jnxStormCtlIfTrafficType	jnxStormCtlEntry 1	<p>Indicates the type of traffic. This object uses the following integer values:</p> <ul style="list-style-type: none"> <li>■ <b>1–Broadcast</b></li> <li>■ <b>2–Multicast</b></li> <li>■ <b>3–Unicast</b></li> </ul>

**Table 220: jnxStormCtlTable** (continued)

Object	Object ID	Description
jnxStormCtlRisingThreshold	jnxStormCtlEntry 2	Specifies the rising threshold value in packets per second unit. Storm control action begins when the traffic exceeds this value.
jnxStormCtlFallingThreshold	jnxStormCtlEntry 3	Specifies the falling threshold value in packets per second unit. Storm control action stops when the traffic drops to this value.
jnxStormCtlAction	jnxStormCtlEntry 4	<p>Specifies the action to be taken when the traffic exceeds the rising threshold. This object returns one of the following integer values:</p> <ul style="list-style-type: none"> <li>■ 1–shutdown: shuts down the port.</li> <li>■ 2–filter: applies a policy filter for the corresponding traffic type on the interface.</li> </ul> <p>The default value is 1, shutdown.</p>

## DHCP Snooping Notification

The jnxSecAccessdsRateLimitCrossed notification, whose object identifier is {jnxSecAccessPortMIBNotifications 1}, is generated when the number of DHCP packets from a source that is not trusted exceeds the jnxSecAccessdsIfRateLimit.

## MAC Limit Exceeded Notification

The jnxSecAccessIfMacLimitExceeded notification, whose object identifier is {jnxSecAccessPortMIBNotifications 2}, is generated when the number of MAC addresses learned by the interface exceeds the maximum number of MAC addresses (jnxSecAccessIfMacLimit) allowed on the interface, and shows the value for jnxSecAccessIfMacLimitExceed.

## Storm Event Notification

The jnxStormEventNotification notification, whose object identifier is {jnxSecAccessPortMIBNotifications 3}, is generated when the traffic on the interface exceeds the jnxStormCtlRisingThreshold value.

## **Part 8**

# **Accounting Options**

- Accounting Options Overview on page 657
- Configuring Accounting Options on page 659
- Summary of Accounting Options Configuration Statements on page 683



## Chapter 80

# Accounting Options Overview

This chapter contains the following topic:

- Accounting Options Overview on page 657

## 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 221 on page 657.

**Table 221: 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.
MIB profile	Collects selected MIB statistics and logs them to a specified file.
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 81

# Configuring Accounting Options

This chapter contains the following topics:

- Accounting Options Configuration on page 659
- Configuring Files on page 662
- Configuring the Interface Profile on page 665
- Configuring the Filter Profile on page 667
- Example: Configuring a Filter Profile on page 669
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 670
- Source Class Usage Options Overview on page 671
- Configuring SCU or DCU on page 672
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 673
- Configuring Class Usage Profiles on page 675
- Configuring the MIB Profile on page 677
- Configuring the Routing Engine Profile on page 679

## Accounting Options Configuration

---

This topic contains the following sections:

- Accounting Options—Full Configuration on page 659
- Minimum Accounting Options Configuration on page 660

### Accounting Options—Full Configuration

To configure accounting options, include the following statements at the [edit accounting-options] hierarchy level:

```
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 number;
        nonpersistent;
        size bytes;
        source-classes time
        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;
}
mib-profile profile-name {
    file filename;
    interval seconds;
    objects-names {
        mib-object-name;
    }
    operation operation-name;
}
routing-engine-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
}

```

By default, accounting options are disabled.

## 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`, `mib-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 number;
    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;
  }
  mib-profile profile-name {
    file filename;
    interval minutes;
    objects-names {
      mib-object-name;
    }
    operation operation-name;
  }
  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 number;
  nonpersistent;
  size bytes;
  start-time time;
  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 Storage Location of the File on page 663
- Configuring the Maximum Size of the File on page 663
- Configuring the Maximum Number of Files on page 663
- Configuring the Start Time for File Transfer on page 664
- Configuring the Transfer Interval of the File on page 664
- Configuring Archive Sites on page 664

### Configuring the Storage Location of the File

On J-series Services Routers, the files are stored by default on the compact flash drive. To configure the storage location of the files in the `mfs/var/log` directory (on DRAM) instead of the `cf/var/log` directory (on the compact flash drive), include the `nonpersistent` statement at the `[edit accounting-options file filename]` hierarchy level:

```
[edit accounting-options file filename]
nonpersistent;
```

This feature is useful for minimizing read/write traffic on the router's compact flash drive.



**NOTE:** If log files for accounting data are stored on DRAM, these files are lost when you reboot the router. Therefore, you should backup these files periodically.

---

### 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 files, include the `files` statement at the `[edit accounting-options file filename]` hierarchy level:

```
[edit accounting-options file filename]
files number;
```

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 *filename* is 3 and the default value is 10.

### Configuring the Start Time for File Transfer

To configure the start time for transferring files, include the **start-time** statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
start-time time;
```

The **start-time** statement specifies a start time for file transfer (YYYY-MM-DD.HH:MM). For example, 10:00 a.m. on January 30, 2007 would be configured as 2007-01-30.10:00.

### 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 **transfer-interval** is 5 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 665
- Configuring the File Information on page 665
- Configuring the Interval on page 666
- Example: Configuring the Interface Profile on page 666

### 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 `file` 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 the `interval` statement 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 {
    xe-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,xe-1/0/0.0,8,134696815,3681534,501088,40723,0,0
if_profile1,976823538,xe-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 on page 668
- Configuring the File Information on page 668
- Configuring the Interval on page 668

## 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 {
}
```

## 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;
```



**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 the `interval` statement 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]
xe-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,xe-1/0/0.0,f3-xe-1/0/0.0-i,r1-xe-1/0/0.0-i,5953,1008257
cust1_profile,995495602,xe-1/0/0.0,f3-xe-1/0/0.0-o,r1-xe-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,xe-1/0/0.0,f3,r1,5953,1008257
cust1_profile,995495632,xe-1/0/0.0,f3,r1,5929,1006481

```

## Source Class Usage Options Overview

---

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.

Class-based filter match conditions are not supported on J-series Services Routers.

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*.

## Configuring SCU or DCU

To configure SCU or DCU, perform the following tasks described in this section:



**NOTE:** We recommend that you stop the network traffic on an interface before you modify the DCU or SCU configuration for that interface. Modifying the DCU or SCU configuration without stopping the traffic might corrupt the DCU or SCU statistics. Before you restart the traffic after modifying the configuration, enter the `clear interfaces statistics` command.

- Creating Prefix Route Filters in a Policy Statement on page 672
- Applying the Policy to the Forwarding Table on page 672
- Enabling Accounting on Inbound and Outbound Interfaces on page 672

### 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 on inbound and outbound interfaces:

```
[edit]
interfaces {
  so-6/1/0 {
    unit 0 {
      family inet;
      accounting {
        destination-class-usage;
        source-class-usage {
```

```

        output;
    }
}
}
}
[edit]
interfaces {
  xe-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 {
  xe-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 Tunnel Interface

To configure source class usage on the virtual loopback tunnel interface, perform the tasks described in the following sections:

- Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC on page 674
- Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface on page 674
- Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface on page 674

### **Example: Configuring a Virtual Loopback Tunnel 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 Tunnel Interface**

Map the VRF instance type to the virtual loopback tunnel 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 Tunnel Interface Out the Source Class Output Interface**

Send traffic received from the virtual loopback tunnel 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 about configuring source class usage on the virtual loopback tunnel 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 675
- Configuring the File Information on page 676
- Configuring the Interval on page 676
- Creating a Class Usage Profile to Collect Source Class Usage Statistics on page 676
- Creating a Class Usage Profile to Collect Destination Class Usage Statistics on page 677

### 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;
```

## 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,xe-1/0/0.0,gold,82,6888  
scu_profile,980313078,xe-1/0/0.0,silver,164,13776  
scu_profile,980313078,xe-1/0/0.0,bronze,0,0  
scu_profile,980313678,xe-1/0/0.0,gold,82,6888  
scu_profile,980313678,xe-1/0/0.0,silver,246,20664  
scu_profile,980313678,xe-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,xe-1/0/0.0,gold,82,6888
dcu_profile,980313078,xe-1/0/0.0,silver,164,13776
dcu_profile,980313078,xe-1/0/0.0,bronze,0,0
dcu_profile,980313678,xe-1/0/0.0,gold,82,6888
dcu_profile,980313678,xe-1/0/0.0,silver,246,20664
dcu_profile,980313678,xe-1/0/0.0,bronze,0,0
...

#FILE CLOSED 976826178 2000-12-14-20:36:18
```

## Configuring the MIB Profile

The MIB profile collects MIB statistics and logs them to a file. The MIB profile specifies the SNMP operation and MIB object names for which statistics are collected.

To configure a MIB profile, include the `mib-profile` statement at the `[edit accounting-options]` hierarchy level:

```
[edit accounting-options]
mib-profile profile-name {
  file filename;
  interval minutes;
  objects-names {
    mib-object-name;
  }
  operation operation-name;
}
```

To configure a MIB profile, perform the tasks described in the following sections:

- Configuring the File Information on page 678
- Configuring the Interval on page 678
- Configuring the MIB Operation on page 678
- Configuring MIB Object Names on page 678
- Example: Configuring a MIB Profile on page 679

## 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 mib-profile profile-name]` hierarchy level:

```
[edit accounting-options mib-profile profile-name]
file filename;
```

You must specify a *filename* for the MIB profile that has already been configured at the `[edit accounting-options]` hierarchy level.

## Configuring the Interval

A MIB 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 mib-profile profile-name]` hierarchy level:

```
[edit accounting-options mib-profile profile-name]
interval;
```

The range for the `interval` statement is 1 through 2880 minutes. The default is 30 minutes.

## Configuring the MIB Operation

A MIB profile must specify the operation that is used to collect MIB statistics. To configure which operation is used to collect MIB statistics, include the `operation` statement at the `[edit accounting options mib-profile profile-name]` hierarchy level:

```
[edit accounting-options mib-profile profile-name]
operation operation-name;
```

You can configure a `get`, `get-next`, or `walk` operation. The default operation is `walk`.

## Configuring MIB Object Names

A MIB profile must specify the MIB objects for which statistics are to be collected. To configure the MIB objects for which statistics are collected, include the `objects-names` statement at the `[edit accounting options mib-profile profile-name]` hierarchy level:

```
[edit accounting-options mib-profile profile-name]
objects-names {
  mib-object-name;
}
```

You can include multiple MIB object names in the configuration.

### Example: Configuring a MIB Profile

Configure a MIB profile:

```
[edit accounting-options]
mib-profile mstatistics {
  file stats;
  interval 60;
  operation walk;
  objects-names {
    ipCidrRouteStatus;
    ifOutOctets;
  }
}
```

## 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 680
- Configuring the File Information on page 680
- Configuring Fields on page 680
- Configuring the File Information on page 680
- Configuring the Interval on page 680
- Example: Configuring a Routing Engine Profile on page 681

## 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 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;
```

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 82

# Summary of Accounting Options Configuration Statements

The following sections explain each of the accounting options configuration statements. The statements are organized alphabetically.

### accounting-options

---

<b>Syntax</b>	accounting-options {...} }
<b>Hierarchy Level</b>	[edit]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Configure options for accounting statistics collection.
<b>Usage Guidelines</b>	See “Configuring Accounting Options” on page 659.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## archive-sites

---

<b>Syntax</b>	<pre>archive-sites {     site-name; }</pre>
<b>Hierarchy Level</b>	[edit accounting-options file <i>filename</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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 <i>router-name_log-filename_timestamp</i> .
<b>Options</b>	<i>site-name</i> —Any valid FTP URL to a destination. For information on how to specify valid FTP URLs, see the <i>JUNOS System Basics Configuration Guide</i> .
<b>Usage Guidelines</b>	See “Configuring Archive Sites” on page 664.
<b>Required Privilege Level</b>	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 675.

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

## counters

---

<b>Syntax</b>	counters { <i>counter-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options filter-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	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 <i>/var/log</i> directory.
<b>Options</b>	<i>counter-name</i> —Name of the counter.
<b>Usage Guidelines</b>	See “Configuring the Counters” on page 668.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## destination-classes

---

<b>Syntax</b>	destination-classes { <i>destination-class-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options class-usage-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the destination classes for which statistics are collected.
<b>Options</b>	<i>destination-class-name</i> —Name of the destination class to include in the source class usage profile.
<b>Usage Guidelines</b>	See “Configuring a Class Usage Profile” on page 675.
<b>Required Privilege Level</b>	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) on page 687
- `fields` (for Routing Engine Profiles) on page 688

### ***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 665.

**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)**

<b>Syntax</b>	fields { <i>field-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options routing-engine-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Statistics to collect in an accounting-data log file for a Routing Engine.
<b>Options</b>	<i>field-name</i> —Name of the field: <ul style="list-style-type: none"> <li>■ <i>cpu-load-1</i>—Average system load over the last 1 minute</li> <li>■ <i>cpu-load-5</i>—Average system load over the last 5 minutes</li> <li>■ <i>cpu-load-15</i>—Average system load over the last 15 minutes</li> <li>■ <i>date</i>—Date, in YYYYMMDD format</li> <li>■ <i>host-name</i>—Hostname for the router</li> <li>■ <i>time-of-day</i>—Time of day, in HHMMSS format</li> <li>■ <i>uptime</i>—Time since last reboot, in seconds</li> </ul>
<b>Usage Guidelines</b>	See “Configuring the Routing Engine Profile” on page 679.
<b>Required Privilege Level</b>	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) on page 689
- file (Configuring a Log File) on page 690

### ***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 mib-profile *profile-name*],  
[edit accounting-options routing-engine-profile *profile-name*]

**Release Information** Statement introduced before JUNOS Release 7.4.  
The [edit accounting-options mib-profile *profile-name*] hierarchy added in JUNOS Release 8.2.

**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 665, “Configuring the Filter Profile” on page 667, “Configuring the MIB Profile” on page 677, and “Configuring the Routing Engine Profile” on page 679.

**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 *number*;  
               nonpersistent;  
               size *bytes*;  
               source-classes *time*;  
               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 the file in which to write the accounting data.

The remaining statements are explained separately.

**Usage Guidelines**    See “Configuring Files” on page 662.

**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 log files 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 662.

**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 667.

**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 665.

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

## interval

---

<b>Syntax</b>	interval <i>minutes</i> ;
<b>Hierarchy Level</b>	[edit accounting-options class-usage-profile <i>profile-name</i> ], [edit accounting-options filter-profile <i>profile-name</i> ], [edit accounting-options interface-profile <i>profile-name</i> ], [edit accounting-options mib-profile <i>profile-name</i> ], [edit accounting-options routing-engine-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4. The [edit accounting-options mib-profile <i>profile-name</i> ] hierarchy level added in JUNOS Release 8.2.
<b>Description</b>	How often statistics are collected for the accounting profile.
<b>Options</b>	<i>minutes</i> —Amount of time between each collection of statistics. <b>Range:</b> 1 through 2880 minutes <b>Default:</b> 30 minutes
<b>Usage Guidelines</b>	See “Configuring the Interface Profile” on page 665, “Configuring the Filter Profile” on page 667, “Configuring the MIB Profile” on page 677, and “Configuring the Routing Engine Profile” on page 679.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.


## mib-profile

---

<b>Syntax</b>	mib-profile <i>profile-name</i> { file <i>filename</i> ; interval <i>minutes</i> ; objects-names { <i>mib-object-name</i> ; } operation <i>operation-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.2.
<b>Description</b>	Create a MIB profile to collect selected MIB statistics and write them to a file in the /var/log directory.
<b>Options</b>	<i>profile-name</i> —Name of the MIB statistics profile.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Configuring the MIB Profile” on page 677.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## nonpersistent

---

<b>Syntax</b>	nonpersistent;
<b>Hierarchy Level</b>	[edit accounting-options file <i>filename</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.3.
<b>Description</b>	For J-series Services Routers only. Stores log files used for accounting data in the mfs/var/log directory (located on DRAM) instead of the cf/var/log directory (located on the compact flash drive). This feature is useful for minimizing read/write traffic on the router’s compact flash drive.
	<b>NOTE:</b> If log files for accounting data are stored on DRAM, these files are lost when you reboot the router. Therefore, you should backup these files periodically.
<b>Usage Guidelines</b>	See “Configuring the Storage Location of the File” on page 663.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## objects-names

---

<b>Syntax</b>	objects-names { <i>mib-object-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options mib-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.2.
<b>Description</b>	Name of the MIB objects for which MIB statistics are collected for an accounting-data log file.
<b>Options</b>	<i>mib-object-name</i> —Name of a MIB object. You can specify more than one MIB object name.
<b>Usage Guidelines</b>	See “Configuring the MIB Profile” on page 677.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## operation

---

<b>Syntax</b>	operation <i>operation-name</i> ;
<b>Hierarchy Level</b>	[edit accounting-options mib-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.2.
<b>Description</b>	Name of the operation used to collect MIB statistics for an accounting-data log file.
<b>Options</b>	<i>operation-name</i> —Name of the operation to use. You can specify a <b>get</b> , <b>get-next</b> , or <b>walk</b> operation. <b>Default:</b> walk
<b>Usage Guidelines</b>	See “Configuring the MIB Profile” on page 677.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## routing-engine-profile

---

<b>Syntax</b>	routing-engine-profile <i>profile-name</i> { fields { <i>field-name</i> ; } file <i>filename</i> ; interval <i>minutes</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Create a Routing Engine profile to collect selected Routing Engine statistics and write them to a file in the <i>/var/log</i> directory.
<b>Options</b>	<i>profile-name</i> —Name of the Routing Engine statistics profile.  The remaining statements are explained separately.
<b>Usage Guidelines</b>	See “Configuring the Routing Engine Profile” on page 679.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## size

---

<b>Syntax</b>	size <i>bytes</i> ;
<b>Hierarchy Level</b>	[edit accounting-options file <i>filename</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Attributes of an accounting-data log file.
<b>Options</b>	<i>bytes</i> —Maximum size of each log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a log file (for example, <b>profilelog</b> ) reaches its maximum size, it is renamed <b>profilelog.0</b> , then <b>profilelog.1</b> , 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. <b>Syntax:</b> x to specify bytes, xk to specify KB, xm to specify MB, xg to specify GB <b>Range:</b> 256 KB through 1 GB
<b>Usage Guidelines</b>	See “Configuring the Maximum Size of the File” on page 663.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## source-classes

---

<b>Syntax</b>	source-classes { <i>source-class-name</i> ; }
<b>Hierarchy Level</b>	[edit accounting-options class-usage-profile <i>profile-name</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Specify the source classes for which statistics are collected.
<b>Options</b>	<i>source-class-name</i> —Name of the source class to include in the class usage profile.
<b>Usage Guidelines</b>	See “Configuring a Class Usage Profile” on page 675.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## start-time

---

<b>Syntax</b>	start-time <i>time</i> ;
<b>Hierarchy Level</b>	[edit accounting-options file <i>filename</i> ]
<b>Release Information</b>	Statement introduced in JUNOS Release 8.2.
<b>Description</b>	Start time for transfer of an accounting-data log file.
<b>Options</b>	<i>time</i> —Start time for file transfer. <b>Syntax:</b> YYYY-MM-DD.HH:MM
<b>Usage Guidelines</b>	See “Configuring the Start Time for File Transfer” on page 664.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

## transfer-interval

---

<b>Syntax</b>	transfer-interval <i>minutes</i> ;
<b>Hierarchy Level</b>	[edit accounting-options file <i>filename</i> ]
<b>Release Information</b>	Statement introduced before JUNOS Release 7.4.
<b>Description</b>	Time the file remains open and receives new statistics before it is closed and transferred to an archive site.
<b>Options</b>	<i>minutes</i> —Time the file remains open and receives new statistics before it is closed and transferred to an archive site. <b>Range:</b> 5 through 2880 minutes <b>Default:</b> 30 minutes
<b>Usage Guidelines</b>	See “Configuring the Transfer Interval of the File” on page 664.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.



## **Part 9**

# **Index**

- Index on page 701
- Index of Statements and Commands on page 711



# Index

## Symbols

#, comments in configuration statements.....	xlvi
( ), in syntax descriptions.....	xlvi
/var/log/mib2d file.....	46
/var/log/snmpd file.....	46
< >, in syntax descriptions.....	xlvi
[ ], in configuration statements.....	xlvi
{ }, in configuration statements.....	xlvi
(pipe), in syntax descriptions.....	xlvi

## A

AAA Objects MIB.....	123, 573
Text Conventions.....	573
Access Authentication Objects MIB.....	123, 577
access authentication traps.....	574
access statement	
usage guidelines.....	62
accounting options	
configuration.....	659
overview.....	657
accounting profiles	
filter.....	667
interface.....	665
MIB.....	677
Routing Engine.....	679
accounting-options statement.....	683
Adaptive Services (AS) PIC.....	461
address statement	
SNMPv3.....	183
usage guidelines.....	71
address-mask statement.....	184
usage guidelines.....	71
agent, SNMP.....	22
agent-address statement.....	165
Alarm MIB.....	123, 539
alarm statement	
RMON.....	239
usage guidelines.....	223
Analyzer MIB.....	123, 637
apasChanStatusTable.....	425
apsChanConfigTable.....	424
apsConfigTable.....	419
apsStatusTable.....	421

archive-sites statement	
accounting.....	684
usage guidelines.....	664
ATM CoS MIB.....	124, 515
ATM MIB.....	124, 523
authentication-md5 statement.....	184
usage guidelines.....	58
authentication-none statement.....	185
usage guidelines.....	59
authentication-password statement.....	185
usage guidelines.....	58
authentication-sha statement.....	186
usage guidelines.....	58
authorization statement.....	166
usage guidelines.....	36

## B

BFD MIB.....	124, 489
notification variables.....	490
BGP4 V2 MIB.....	124, 389
braces, in configuration statements.....	xlvi
brackets	
angle, in syntax descriptions.....	xlvi
square, in configuration statements.....	xlvi

## C

categories statement.....	166
usage guidelines.....	41
Chassis Definitions for Router Model MIB.....	124, 380
Chassis Forwarding MIB.....	124, 475
Chassis MIB.....	124
jnxBoxAnatomy.....	295
jnxBoxKernelMemoryUsedPercent.....	375
jnxMIBs.....	295
jnxTraps.....	375
overview.....	295
Class 1 MIB objects.....	115
Class 2 MIB objects.....	119
Class 3 MIB objects.....	120
Class 4 MIB objects.....	121
Class-of-Service MIB.....	124
class-usage-profile statement.....	685
usage guidelines.....	675

client list	
adding to SNMP community.....	37
client-list statement.....	167
usage guidelines.....	37
client-list-name statement.....	167
usage guidelines.....	37
clients statement.....	168
usage guidelines.....	36
comments, in configuration statements.....	xlvi
commit-delay statement.....	168
usage guidelines.....	35
community statement	
RMON.....	240
usage guidelines.....	226
SNMP.....	169
usage guidelines.....	36
community string, SNMP.....	36
community-name statement.....	187
usage guidelines.....	81
Configuration Management MIB.....	124
contact statement.....	170
usage guidelines.....	34
conventions	
text and syntax.....	xliv
CoS	
measuring.....	282
MIB.....	124
counters statement.....	686
curly braces, in configuration statements.....	xlvi
customer support.....	liii
contacting JTAC.....	liii

**D**

DCU, Destination Class Usage <i>See</i> Destination Class	
Usage MIB	
description statement	
RMON.....	240
usage guidelines (alarms).....	223
usage guidelines (events).....	226
SNMP.....	170
usage guidelines.....	34
Destination Class Usage MIB.....	124, 387
destination-classes statement.....	686
usage guidelines.....	675
destination-port statement	
SNMP.....	171
usage guidelines.....	41
DFC, Dynamic Flow Capture <i>See</i> Dynamic Flow	
Capture MIB	
DNS Objects MIB.....	125, 579
documentation set	
comments on.....	liii
dropped traffic	
measuring.....	285

Dynamic Flow Capture MIB.....	125, 381, 467
notification definitions.....	472
notification variables.....	471, 488

**E**

engine-id statement	
SNMPv3.....	188
usage guidelines.....	56
enterprise-specific MIBs, listed.....	123
enterprise-specific traps, SNMP	
unsupported.....	140
version 1.....	131
version 2.....	135
Ethernet MAC MIB.....	125, 437
Event MIB.....	125, 487
event statement.....	241
usage guidelines.....	226
EX-series Ethernet switches	
enterprise-specific traps.....	139
MIB objects.....	383
standard traps.....	159
Experimental MIB.....	125

**F**

falling-event-index statement.....	241
usage guidelines.....	223
falling-threshold statement	
health monitor.....	257
usage guidelines.....	253
RMON.....	242
falling-threshold-interval statement	
RMON.....	242
usage guidelines.....	224
fields statement	
for interface profiles.....	687
usage guidelines.....	665
for Routing Engine profiles.....	688
usage guidelines.....	680
file statement	
accounting (associating with profile).....	689
usage guidelines (filter profile).....	668
usage guidelines (interface profile).....	665
usage guidelines (MIB profile).....	678
usage guidelines (Routing Engine	
profile).....	680
accounting (configuring log file).....	690
usage guidelines.....	662
files statement.....	690
filter profile.....	667
filter-duplicates statement.....	171
usage guidelines.....	35
filter-profile statement.....	691
usage guidelines.....	667
filtering get SNMP requests.....	35

Firewall MIB.....125  
 Flow Collection Services MIB.....125, 457, 483  
 font conventions.....xliv

**G**

Get requests, SNMP.....19  
 group statement  
     SNMPv3 (for access privileges).....189  
         usage guidelines.....67  
     SNMPv3 (for configuring).....189  
         usage guidelines.....63

**H**

health metrics of network.....274  
 health-monitor statement.....258  
     usage guidelines.....253  
 Host Resources MIB.....125, 549

**I**

icons defined, notice.....xliv  
 ifChassisTable.....441  
 ILM1.....15  
 inform-retry-count statement.....190  
     usage guidelines.....78  
 inform-timeout statement.....190  
     usage guidelines.....78  
 informs SNMP *See* SNMP informs  
 integrated local management interface *See* ILM1  
 Interface MIB.....125, 439  
 interface profile.....665  
 interface statement  
     SNMP.....172  
         usage guidelines.....44  
 interface-profile statement.....692  
     usage guidelines.....665  
 interfaces limiting SNMP access.....44  
 interval statement  
     accounting.....693  
         usage guidelines (filter profile).....668  
         usage guidelines (interface profile).....666  
         usage guidelines (MIB profile).....678  
         usage guidelines (Routing Engine  
         profile).....680  
     health monitor.....258  
         usage guidelines.....254  
     RMON.....243  
         usage guidelines.....224  
 IP Forward MIB.....125, 513  
 IPsec Generic Flow Monitoring Object MIB.....126, 581  
     Text Conventions.....582  
 IPsec Monitoring MIB.....125, 429  
 IPsec Phase 1 IKE Tunnel Table.....586  
 IPsec Phase 2 IKE Tunnel Table.....589

IPsec Phase 2 Security Association Table.....592  
 IPsec VPN Objects MIB.....126, 595  
     Text Conventions.....595  
 IPv4 MIB.....126, 537  
 IPv6 and ICMPv6 MIB.....126  
 IPv6 SNMP community string.....36

**J**

jnxBfdSessTable.....489  
 jnxBgpM2PrefixCountersTable.....389  
 jnxBoxAnatomy MIB.....295  
 jnxBoxKernelMemoryUsedPercent.....375  
 jnxCollFileTable.....459  
 jnxCollGlobalStats.....457  
 jnxCollPicIfTable.....458  
 jnxContainersTable  
     M160 router.....299  
     M20 router.....299  
     M40 router.....298  
     M40e router.....302  
     M5 router.....301  
     T320 router.....302  
     T640 routing node.....301  
 jnxContentsTable  
     M20 router.....304  
     T320 router.....311  
     T640 routing node.....307  
 jnxCosInvQstatTable.....509  
 jnxDCUsTable.....387  
 jnxDcuStatsTable.....388  
 jnxDfcCDTable.....471  
 jnxDfcCSTable.....467  
 jnxEventAvTable.....487  
 jnxExperiment root branch.....292  
 jnxFilledTable.....318  
     M20 router.....318  
     T320 router.....324  
     T640 routing node.....321  
 jnxFruTable.....340  
     M10 router.....343  
     M160 router.....349  
     M20 router.....346  
     M40 router.....356  
     M40e router.....361  
     T640 routing node.....366  
 jnxFwddProcess.....475  
 jnxIfTable.....439  
 jnxIkeTunnelTable.....429  
 jnxIPSecSaTable.....434  
 jnxIPSecTunnelTable.....432  
 jnxJsAuthNotifications.....578  
 jnxJsAuthTrapVars.....578  
 jnxJsDnsProxyDataObjects.....579  
 jnxJsFwAuthStats.....577  
 jnxJsIfMonTable.....609

jnxJsIpSecTunnelTable.....	596
jnxJsLoadedCaCertTable.....	613
jnxJsLoadedLocalCertTable.....	614
jnxJsNatIfSrcPoolPortTable.....	601
jnxJsPolicyStatsTable.....	605
jnxJsScreenMonTable.....	615
jnxLEDTTable.....	314
M20 router.....	316
T320 router.....	317
T640 routing node.....	316
jnxMacStatsTable.....	437
jnxMibs root branch.....	290
jnxOperatingTable.....	327
M20 router.....	330
T320 router.....	333
T640 routing node.....	330
jnxPfeNotifyGfTable.....	483
jnxPfeNotifyTypeTable.....	485
jnxPingCtlTable.....	391
jnxPingLastTestResultTable.....	400
jnxpingProbeHistoryTable.....	398
jnxPingResultsTable.....	395
jnxPMonFlowTable.....	415
jnxProducts root branch.....	289
jnxRedundancyTable.....	335
M20 router.....	337
T320 router.....	339
T640 routing node.....	338
jnxRmonAlarmGetFailure.....	409
jnxRmonAlarmTable.....	230, 407
jnxRmonGetOk.....	409
jnxRpfStatsTable.....	411
jnxScuStatsTable.....	413, 509
jnxServices root branch.....	289
jnxSonetAlarmsTable.....	417
jnxSpSvcSetIfTable.....	463
jnxSpSvcSetSvcTypeTable.....	463
jnxSpSvcSetTable.....	461
jnxSyslogAvTable.....	479
jnxSyslogTable.....	477
jnxTraceRouteCtlTable.....	405
jnxTraps root branch.....	292
jnxUserAAAServerName.....	574
jnxUserAAASatTable.....	574
jnxUtilCounter32Table.....	569
jnxUtilCounter64Table.....	570
jnxUtilIntegerTable.....	570
jnxUtilStringTable.....	571
jnxUtilUIntTable.....	570
jnxVpnIfTable.....	447
jnxVpnInfo.....	445
jnxVpnPwTable.....	450
jnxVpnRTTable.....	455
jnxVpnTable.....	446
Juniper Networks MIB objects.....	111

**K**

key performance indicators.....	264
---------------------------------	-----

**L**

L2ALD MIB.....	126, 567
L2CP features MIB.....	126
L2TP MIB.....	126, 491
Layer 2 Control Protocol	
MIB.....	551
LDP	
MIB.....	631
traps.....	139, 631
disabling.....	139
LDP MIB.....	126
local-engine statement.....	191
location statement	
SNMP.....	172
usage guidelines.....	34
logical-system statement.....	173

**M**

M120 router	
MIB objects.....	381
Management Information Base <i>See</i> MIBs	
Management Information MIB	
jnxMibs.....	290
jnxProducts.....	289
jnxServices.....	289
jnxTraps.....	292
manuals	
comments on.....	liii
master agent, SNMP.....	22
measurement tests	
proxy ping.....	272
message-processing-model statement.....	192
usage guidelines.....	75
MIB object classes.....	106
MIB profile.....	677
mib-profile statement.....	694
usage guidelines.....	677
MIBs	
AAA Objects.....	123, 573
Access Authentication Objects.....	123, 577
Alarm.....	123, 539
ATM.....	124, 523
ATM CoS.....	124, 515
BFD.....	124, 489
BGP4 V2.....	124, 389
Chassis.....	124, 295, 375
Chassis Definitions for Router Model.....	124, 380
Chassis Forwarding.....	124, 475
Class-of-Service.....	124
Configuration Management.....	124
Destination Class Usage.....	124, 387

DNS Objects.....125, 579  
 Dynamic Flow Capture.....125, 381, 467  
 enterprise-specific, listed.....123  
 Ethernet MAC.....125, 437  
 Event.....125, 487  
 EX-series  
     Analyzer.....123, 637  
     PAE Extension.....127, 647  
     Secure Access Port.....128, 651  
     SMI.....635  
     Structure of Management Information  
         Base.....129, 635  
         Virtual Chassis.....129, 645  
         VLAN.....129, 641  
 Experimental.....125  
 Firewall.....125  
 Flow Collection Services.....125, 457, 483  
 Host Resources.....125, 549  
 Interface.....125, 439  
 IP Forward.....125, 513  
 IPSec Generic Flow Monitoring Object .....126, 581  
 IPSec Monitoring.....125, 429  
 IPSec VPN Objects.....126, 595  
 IPv4.....126, 537  
 IPv6 and ICMPv6.....126  
 L2ALD.....126, 567  
 L2CP Features.....126  
 L2TP.....126, 491  
 Layer 2 Control Protocol.....551  
 LDP.....126, 631  
 Management Information MIB  
     jnxMibs.....290  
     jnxProducts.....289  
     jnxServices.....289  
     jnxTraps.....292  
 MIMSTP.....126, 553  
 MPLS.....126, 543  
 MPLS LDP.....481  
 Multicast.....26, 29, 30  
 NAT Objects.....127, 599  
 OSPF.....24  
 Packet Forwarding Engine.....127, 483  
 Passive Monitoring.....127, 415  
 Ping.....127  
     interpretation of.....391  
     use in ping test.....90  
     view configuration example, SNMP.....45  
 Policy Objects.....127, 603  
 Reverse-Path-Forwarding.....128, 411  
 RMON Events and Alarms .....128, 407  
 RPM.....127, 501  
 RSVP.....541  
 RSVP TE.....128  
 Security Interface Extension Objects.....128, 609  
 Security Screening Objects.....128, 615  
 Services PIC.....128, 461

SONET APS.....128, 419  
 SONET/SDH Interface Management.....128, 417  
 Source Class Usage.....128, 413, 509  
 standards documents.....23  
 Structure of Management Information.....129, 289  
     JUNOS software with enhanced services,  
         for.....129  
     System Log.....129, 477  
     Traceroute.....129, 405  
     Utility.....129, 569  
     views  
         SNMP.....45  
     VPN.....129, 445  
     VPN Certificate Objects.....129, 613  
 MIMSTP  
     MIB.....126, 553  
 minimum accounting options configuration.....660  
 monitoring  
     service quality.....263  
 MPLS  
     enterprise-specific traps.....547  
     MIB.....126  
     standard traps.....152  
 MPLS LDP MIB.....481  
 MPLS MIB.....543  
 Multicast MIB.....26, 29, 30  
 MX240 Ethernet Services Router  
     MIB objects.....383  
 MX480 Ethernet Services Router  
     MIB objects.....382  
 MX960 Ethernet Services Router  
     MIB objects.....382

## N

name statement.....173  
     usage guidelines.....35  
 NAT Objects MIB.....127, 599  
 NAT trap definitions.....601  
 Network Address Translation Objects MIB *See* NAT  
     Objects MIB  
 network health  
     measuring.....274  
 network performance  
     measuring.....280  
 nonpersistent statement.....694  
     accounting  
         usage guidelines.....663  
 nonvolatile statement.....174  
 notice icons defined.....xliv  
 notify statement.....192  
     usage guidelines.....69

notify-filter statement	
for applying to target.....	193
usage guidelines.....	74
for configuring.....	193
usage guidelines.....	70
notify-view statement.....	194
usage guidelines.....	65
number of IKE Tunnels currently active .....	585
nxContainersTable	
M10 router.....	300

## O

objects-names statement.....	695
for Routing Engine profiles	
usage guidelines.....	678
oid statement	
SNMP.....	174
usage guidelines.....	45
SNMPv3.....	194
usage guidelines.....	70
operation statement.....	695
for MIB profiles	
usage guidelines.....	678
opsfVirtIfStateChange SNMP trap.....	153
OSPF MIB.....	24
ospfIfAuthFailure SNMP trap.....	154
ospfIfConfigError SNMP trap.....	154
ospfIfRxBadPacket SNMP trap.....	155
ospfIfStateChange SNMP trap.....	153
ospfMaxAgeLsa SNMP trap.....	156
ospfNbrStateChange SNMP trap.....	153
ospfTxRetransmit SNMP trap.....	155
ospfVirtIfAuthFailure SNMP trap.....	155
ospfVirtIfConfigError SNMP trap.....	154
ospfVirtIfRxBadPacket SNMP trap.....	155
ospfVirtNbrStateChange SNMP trap.....	153
ospfVirtTxRetransmit SNMP trap.....	156

## P

Packet Forwarding Engine MIB.....	127, 483
PAE Extension MIB.....	127, 647
parameters statement.....	195
usage guidelines.....	74
parentheses, in syntax descriptions.....	xlvi
Passive Monitoring MIB.....	127, 415
performance indicators.....	264
performance, monitoring.....	280
Ping MIB.....	127
interpretation of.....	391
use in ping test.....	90
view configuration example	
SNMP.....	45
pingCtlTable.....	272
pingProbeHistoryTable.....	95

Policy Objects MIB.....	127, 603
port statement	
SNMPv3.....	195
usage guidelines.....	72
prefix list	
adding to SNMP community.....	37
privacy-3des statement.....	196
usage guidelines.....	60
privacy-aes128 statement.....	197
usage guidelines.....	59
privacy-des statement.....	198
usage guidelines.....	60
privacy-none statement.....	198
usage guidelines.....	60
privacy-password statement.....	199
usage guidelines	
for 3DES algorithm.....	60
for AES algorithm.....	59
for DES algorithm.....	60
profiles, accounting	
filter.....	667
interface.....	665
MIB.....	677
Routing Engine.....	679
proxy ping	
measurement tests.....	272

## R

read-view statement.....	199
usage guidelines.....	65
real-time performance monitoring	
in service provider networks.....	272
redundant adaptive services interfaces (rsp).....	465
remote operations MIBs.....	89
remote-engine statement.....	200
request snmp spoof-trap command.....	141, 164
request-type statement.....	243
RMON	
usage guidelines.....	225
Reverse-Path-Forwarding MIB.....	128, 411
rising-event-index statement.....	244
usage guidelines.....	223
rising-threshold statement	
health monitor.....	259
RMON.....	244
RMON alarm entries.....	222
RMON alarms.....	229, 267
RMON event entries.....	226
RMON events.....	234, 266
RMON Events and Alarms MIB.....	128, 407
rmon statement.....	245
usage guidelines.....	266
Routing Engine profile.....	679



routing instances	
access lists	
configuring.....	110
SNMP	
enabling access.....	108
identifying.....	107
specifying.....	108
routing-engine-profile statement.....	696
usage guidelines.....	679
routing-instance statement	
SNMP.....	175
SNMPv3.....	201
usage guidelines.....	72
RPM MIB.....	127, 501
RSVP MIB.....	541
RSVP TE MIB.....	128

## S

sample-type statement.....	245
usage guidelines	
for alarms.....	225
for events.....	226
SCU, Source Class Usage <i>See</i> Source Class Usage MIB	
Secure Access Port.....	651
Secure Access Port MIB.....	128
Security Interface Extension Objects MIB.....	128, 609
Security Policy Table.....	603
Security Screening Objects MIB.....	128, 615
security-level statement	
for access privileges.....	202
usage guidelines.....	63
for SNMP notifications.....	202
usage guidelines.....	75
security-model statement	
for access privileges.....	203
usage guidelines.....	63
for groups.....	203
usage guidelines.....	66
for SNMP notifications.....	204
usage guidelines.....	75
security-name statement.....	205
for community string.....	205
for security group.....	206
usage guidelines.....	67
for SNMP notifications.....	206
usage guidelines.....	76
security-to-group statement.....	207
usage guidelines.....	62
service quality	
monitoring.....	263
Services PIC MIB.....	128, 461
traps.....	464
Set requests, SNMP.....	19
size statement	
accounting.....	696
usage guidelines.....	663
SMI MIB for EX-series.....	635
SNMP	
adding client lists and prefix lists.....	37
agent.....	19, 22
architecture.....	19
commit delay timer.....	35
community string.....	36
configuration	
version 3.....	53, 54
versions 1 and 2.....	32
enterprise-specific traps <i>See</i> SNMP traps	
filtering duplicate requests.....	35
limiting interface access.....	44
logging, enabling.....	90
manager.....	19
master agent.....	22
MIB views.....	45
remote operations.....	87
spoofing traps.....	141, 164
standard traps <i>See</i> SNMP traps	
standards documents.....	22
subagent.....	22
system contact.....	34
system description.....	34
system location.....	34, 172
system name.....	35
tracing operations.....	46
trap groups.....	41
trap notification for remote operations.....	89
trap options.....	39
views, setting.....	88
SNMP informs.....	76
snmp statement.....	175
usage guidelines	
SNMPv1 and SNMPv2.....	32
SNMPv3.....	53, 54
SNMP traps.....	20
enterprise-specific	
version 1.....	131
version 2.....	135
EX-series Ethernet switches.....	139, 159
MAC limit.....	139
MX960 Ethernet Services Router.....	139
source address configuration.....	40
spoofing.....	141, 164
standard	
version 1.....	143
version 2.....	149
system logging severity levels.....	21
unsupported.....	140, 160
snmp-community statement.....	207

SNMPv1	
Ping Traps MIB.....	146
standard traps.....	145
Traceroute Traps MIB.....	147
VRRP Traps MIB.....	148
SNMPv2	
MPLS traps.....	152
OSPF Traps MIB.....	153
Passive Monitoring Traps MIB.....	41
Ping Traps MIB.....	157
standard traps.....	151
Traceroute Traps MIB.....	158
SNMPv3	
authentication, configuring.....	58
informs, configuring.....	76
local engine ID, configuring.....	56
minimum configuration.....	55
SONET APS MIB.....	128
SONET Automatic Protection Switching MIB.....	419
SONET/SDH Interface Management MIB.....	128, 417
Source Class Usage MIB.....	128, 413, 509
Source NAT Table.....	599
source-address statement.....	176
usage guidelines.....	40
source-classes statement.....	697
usage guidelines.....	675
SRX 5600 Services Gateway	
MIB objects.....	385
SRX 5800 Services Gateway	
MIB objects.....	385
standard traps, SNMP	
version 1.....	143
version 2.....	149
standards documents	
SNMP and MIBs.....	23
start-time statement	
accounting.....	697
usage guidelines.....	664
startup-alarm statement.....	246
usage guidelines.....	226
Structure of Management Information Base MIB	
for EX-series.....	129, 635
Structure of Management Information MIB.....	129, 289
JUNOS software with enhanced services,	
for.....	129
subagent, SNMP.....	22
support, technical <i>See</i> technical support	
syntax conventions.....	xliv
sysContact object, MIB II.....	34
sysDescription object, MIB II.....	34
sysLocation object, MIB II.....	34
syslog-subtag statement.....	246
usage guidelines.....	226
sysName object, MIB II.....	35
system contact, SNMP.....	34
system description, SNMP.....	34
system location, SNMP.....	34, 172
system log messages	
as basis for SNMP traps.....	140
System Log MIB.....	129, 477
system logging severity levels, SNMP traps.....	21
system name, SNMP.....	35
<b>T</b>	
tag statement.....	208
SNMPv3	
usage guidelines.....	82
usage guidelines.....	69
tag-list statement.....	208
usage guidelines.....	72
target-address statement.....	209
usage guidelines.....	70
target-parameters statement.....	210
usage guidelines.....	74
targets statement.....	176
usage guidelines.....	41
technical support	
contacting JTAC.....	liii
traceoptions statement.....	177
SNMP	
usage guidelines.....	46
Traceroute MIB.....	97, 129, 405
traceRouteHopsTable.....	101
tracing operations	
SNMP.....	46
transfer-interval statement	
accounting.....	698
usage guidelines.....	664
trap groups, SNMP.....	41
trap notification for SNMP remote operations.....	89
trap-group statement.....	179
usage guidelines.....	41
trap-options statement.....	180
usage guidelines.....	39
traps.....	135
definition.....	20
LDP.....	139, 631
MPLS, enterprise-specific.....	547
Services PIC MIB.....	464
SNMP version 1 traps	
enterprise-specific.....	131
standard.....	143
SNMP version 2 traps	
enterprise-specific.....	135
standard.....	149
spoofing SNMP traps.....	141, 164
unsupported.....	140, 160
<i>See also</i> SNMP traps	
type statement.....	247
usage guidelines.....	69

**U**

unsupported enterprise-specific SNMP traps.....	140
unsupported standard SNMP traps.....	160
user statement	
SNMPv3.....	211
usm statement.....	212
Utility MIB.....	129, 569

**V**

v3 statement.....	214
usage guidelines.....	53, 54
vacm statement.....	216
usage guidelines.....	62
var/log/mib2d file.....	46
var/log/snmpd file.....	46
variable statement.....	247
usage guidelines.....	226
variable-length string indexes.....	89
version statement	
SNMP.....	180
usage guidelines.....	41
view statement	
SNMP (associating with community).....	181
usage guidelines.....	36
SNMP (configuring MIB view).....	182
usage guidelines.....	45
SNMPv3.....	217
views, MIB	
SNMP.....	45, 88
Virtual Chassis MIB.....	129, 645
VLAN MIB.....	129, 641
VPN Certificate Objects MIB.....	129, 613
VPN MIB.....	129, 445

**W**

warmStart SNMP trap.....	145
write-view statement.....	218
usage guidelines.....	65



# Index of Statements and Commands

## A

accounting-options statement.....	683
address statement	
SNMPv3.....	183
address-mask statement.....	184
agent-address statement.....	165
alarm statement	
RMON.....	239
archive-sites statement	
accounting.....	684
authentication-md5 statement.....	184
authentication-none statement.....	185
authentication-password statement.....	185
authentication-sha statement.....	186
authorization statement.....	166

## C

categories statement.....	166
class-usage-profile statement.....	685
client-list statement.....	167
client-list-name statement.....	167
clients statement.....	168
commit-delay statement.....	168
community statement	
RMON.....	240
SNMP.....	169
community-name statement.....	187
contact statement.....	170
counters statement.....	686

## D

description statement	
RMON.....	240
SNMP.....	170
destination-classes statement.....	686
destination-port statement	
SNMP.....	171

## E

engine-id statement	
SNMPv3.....	188

event statement.....	241
----------------------	-----

## F

falling-event-index statement.....	241
falling-threshold statement	
health monitor.....	257
RMON.....	242
falling-threshold-interval statement	
RMON.....	242
fields statement	
for interface profiles.....	687
for Routing Engine profiles.....	688
file statement	
accounting (associating with profile).....	689
accounting (configuring log file).....	690
files statement.....	690
filter-duplicates statement.....	171
filter-profile statement.....	691

## G

group statement	
SNMPv3 (for access privileges).....	189
SNMPv3 (for configuring).....	189

## H

health-monitor statement.....	258
-------------------------------	-----

## I

inform-retry-count statement.....	190
inform-timeout statement.....	190
interface statement	
SNMP.....	172
interface-profile statement.....	692
interval statement	
accounting.....	693
health monitor.....	258
RMON.....	243

## L

local-engine statement.....	191
-----------------------------	-----

location statement	
SNMP.....	172
logical-system statement.....	173

**M**

message-processing-model statement.....	192
mib-profile statement.....	694

**N**

name statement.....	173
nonpersistent statement.....	694
nonvolatile statement.....	174
notify statement.....	192
notify-filter statement	
for applying to target.....	193
for configuring.....	193
notify-view statement.....	194

**O**

objects-names statement.....	695
oid statement	
SNMP.....	174
SNMPv3.....	194
operation statement.....	695

**P**

parameters statement.....	195
port statement	
SNMPv3.....	195
privacy-3des statement.....	196
privacy-aes128 statement.....	197
privacy-des statement.....	198
privacy-none statement.....	198
privacy-password statement.....	199

**R**

read-view statement.....	199
remote-engine statement.....	200
request snmp spoof-trap command.....	141, 164
request-type statement.....	243
rising-event-index statement.....	244
rising-threshold statement	
health monitor.....	259
RMON.....	244
rmon statement.....	245
routing-engine-profile statement.....	696
routing-instance statement	
SNMP.....	175
SNMPv3.....	201

**S**

sample-type statement.....	245
security-level statement	
for access privileges.....	202
for SNMP notifications.....	202
security-model statement	
for access privileges.....	203
for groups.....	203
for SNMP notifications.....	204
security-name statement.....	205
for community string.....	205
for security group.....	206
for SNMP notifications.....	206
security-to-group statement.....	207
size statement	
accounting.....	696
snmp statement.....	175
snmp-community statement.....	207
source-address statement.....	176
source-classes statement.....	697
start-time statement	
accounting.....	697
startup-alarm statement.....	246
syslog-subtag statement.....	246

**T**

tag statement.....	208
tag-list statement.....	208
target-address statement.....	209
target-parameters statement.....	210
targets statement.....	176
traceoptions statement.....	177
transfer-interval statement	
accounting.....	698
trap-group statement.....	179
trap-options statement.....	180
type statement.....	247

**U**

user statement	
SNMPv3.....	211
usm statement.....	212

**V**

v3 statement.....	214
vacm statement.....	216
variable statement.....	247
version statement	
SNMP.....	180
view statement	
SNMP (associating with community).....	181
SNMP (configuring MIB view).....	182
SNMPv3.....	217

**W**

write-view statement.....218

