



Junos[®] OS

Network Management Configuration Guide

Release
11.2



Published: 2011-05-17

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

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

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

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® OS Network Management Configuration Guide

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

March 2011—B1 Junos OS 11.2

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 OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

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	xxiii
Part 1	Network Management Introduction	
Chapter 1	Network Management Overview	3
Chapter 2	Complete Network Management Configuration Statements	7
Part 2	Integrated Local Management Interface	
Chapter 3	Integrated Local Management Interface Overview	15
Part 3	Simple Network Management Protocol (SNMP)	
Chapter 4	SNMP Overview	19
Chapter 5	Configuring SNMP	23
Chapter 6	SNMPv3 Overview	47
Chapter 7	Configuring SNMPv3	49
Chapter 8	SNMP Remote Operations	85
Chapter 9	SNMP Support for Routing Instances	105
Chapter 10	Understanding the Junos OS MIB Support	123
Chapter 11	Summary of SNMP Configuration Statements	127
Chapter 12	Summary of SNMPv3 Configuration Statements	151
Part 4	RMON Alarms and Events	
Chapter 13	Configuring RMON Alarms and Events	195
Chapter 14	Monitoring RMON Alarms and Events	203
Chapter 15	Summary of RMON Alarm and Event Configuration Statements	213
Part 5	Health Monitoring	
Chapter 16	Configuring Health Monitoring	225
Chapter 17	Summary of Health Monitoring Configuration Statements	229
Part 6	Monitoring Service Quality	
Chapter 18	Monitoring Service Quality in Service Provider Networks	235
Part 7	Accounting Options	
Chapter 19	Accounting Options Overview	263

Chapter 20	Configuring Accounting Options	265
Chapter 21	Summary of Accounting Options Configuration Statements	289
Part 8	Index	
	Index	307
	Index of Statements and Commands	313

Table of Contents

	About This Guide	xxiii
	Junos OS Documentation and Release Notes	xxiii
	Objectives	xxiv
	Audience	xxiv
	Supported Platforms	xxiv
	Using the Indexes	xxv
	Using the Examples in This Manual	xxv
	Merging a Full Example	xxv
	Merging a Snippet	xxvi
	Documentation Conventions	xxvi
	Documentation Feedback	xxviii
	Requesting Technical Support	xxviii
	Self-Help Online Tools and Resources	xxix
	Opening a Case with JTAC	xxix
Part 1	Network Management Introduction	
Chapter 1	Network Management Overview	3
	Understanding Device Management Functions in Junos OS	3
Chapter 2	Complete Network Management Configuration Statements	7
	Configuration Statements at the [edit accounting-options] Hierarchy Level	7
	Configuration Statements at the [edit snmp] Hierarchy Level	8
Part 2	Integrated Local Management Interface	
Chapter 3	Integrated Local Management Interface Overview	15
	Understanding the Integrated Local Management Interface	15
Part 3	Simple Network Management Protocol (SNMP)	
Chapter 4	SNMP Overview	19
	Understanding the SNMP Implementation in Junos OS	19
	SNMP Architecture	19
	SNMP MIBs	20
	SNMP Traps and Informs	20
	Junos OS SNMP Agent Features	22
Chapter 5	Configuring SNMP	23
	Configuring SNMP on a Device Running Junos OS	24
	Configuring the System Contact on a Device Running Junos OS	26
	Configuring the System Location for a Device Running Junos OS	26

	Configuring the System Description on a Device Running Junos OS	27
	Filtering Duplicate SNMP Requests	27
	Configuring the Commit Delay Timer	28
	Configuring the System Name	28
	Configuring the SNMP Community String	29
	Examples: Configuring the SNMP Community String	30
	Adding a Group of Clients to an SNMP Community	30
	Configuring SNMP Trap Options and Groups on a Device Running Junos OS	32
	Configuring SNMP Trap Options	33
	Configuring the Source Address for SNMP Traps	33
	Configuring the Agent Address for SNMP Traps	35
	Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps	36
	Configuring SNMP Trap Groups	36
	Example: Configuring SNMP Trap Groups	38
	Configuring the Interfaces on Which SNMP Requests Can Be Accepted	39
	Example: Configuring Secured Access List Checking	39
	Filtering Interface Information Out of SNMP Get and GetNext Output	40
	Configuring MIB Views	40
	Example: Ping Proxy MIB	41
	Tracing SNMP Activity on a Device Running Junos OS	42
	Configuring the Number and Size of SNMP Log Files	43
	Configuring Access to the Log File	43
	Configuring a Regular Expression for Lines to Be Logged	44
	Configuring the Trace Operations	44
	Example: Tracing SNMP Activity	45
	Configuring the Local Engine ID	46
Chapter 6	SNMPv3 Overview	47
	SNMPv3 Overview	47
Chapter 7	Configuring SNMPv3	49
	Complete SNMPv3 Configuration Statements	50
	Minimum SNMPv3 Configuration on a Device Running Junos OS	52
	Configuring the Local Engine ID	53
	Creating SNMPv3 Users	54
	Configuring the SNMPv3 Authentication Type	55
	Configuring MD5 Authentication	55
	Configuring SHA Authentication	55
	Configuring No Authentication	56
	Configuring the Encryption Type	56
	Configuring the Advanced Encryption Standard Algorithm	56
	Configuring the Data Encryption Algorithm	57
	Configuring Triple DES	57
	Configuring No Encryption	58
	Defining Access Privileges for an SNMP Group	58
	Configuring the Access Privileges Granted to a Group	59
	Configuring the Group	59
	Configuring the Security Model	60

Configuring the Security Level	60
Associating MIB Views with an SNMP User Group	60
Configuring the Notify View	61
Configuring the Read View	61
Configuring the Write View	62
Example: Access Privilege Configuration	62
Assigning Security Model and Security Name to a Group	63
Configuring the Security Model	63
Assigning Security Names to Groups	64
Configuring the Group	64
Example: Security Group Configuration	64
Configuring SNMPv3 Traps on a Device Running Junos OS	65
Configuring the SNMPv3 Trap Notification	66
Example: Configuring SNMPv3 Trap Notification	67
Configuring the Trap Notification Filter	67
Configuring the Trap Target Address	68
Configuring the Address	69
Configuring the Address Mask	69
Configuring the Port	70
Configuring the Routing Instance	70
Configuring the Trap Target Address	70
Applying Target Parameters	71
Example: Configuring the Tag List	71
Defining and Configuring the Trap Target Parameters	72
Applying the Trap Notification Filter	73
Configuring the Target Parameters	73
Configuring the Message Processing Model	73
Configuring the Security Model	73
Configuring the Security Level	74
Configuring the Security Name	74
Configuring SNMP Informs	75
Configuring the Remote Engine and Remote User	76
Example: Configuring the Remote Engine ID and Remote Users	77
Configuring the Inform Notification Type and Target Address	77
Example: Configuring the Inform Notification Type and Target Address	79
Configuring the SNMPv3 Community	79
Configuring the Community Name	80
Configuring the Security Names	81
Configuring the Tag	81
Example: SNMPv3 Community Configuration	81
Example: SNMPv3 Configuration	82
Chapter 8	
SNMP Remote Operations	85
SNMP Remote Operations Overview	85
SNMP Remote Operation Requirements	86
Setting SNMP Views	86
Example: Setting SNMP Views	86
Setting Trap Notification for Remote Operations	87
Example: Setting Trap Notification for Remote Operations	87

	Using Variable-Length String Indexes	87
	Example: Set Variable-Length String Indexes	87
	Enabling Logging	88
	Using the Ping MIB for Remote Monitoring Devices Running Junos OS	88
	Starting a Ping Test	88
	Using Multiple Set Protocol Data Units (PDUs)	89
	Using a Single Set PDU	89
	Monitoring a Running Ping Test	90
	pingResultsTable	90
	pingProbeHistoryTable	91
	Generating Traps	92
	Gathering Ping Test Results	93
	Stopping a Ping Test	94
	Interpreting Ping Variables	94
	Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS	95
	Starting a Traceroute Test	96
	Using Multiple Set PDUs	96
	Using a Single Set PDU	96
	Monitoring a Running Traceroute Test	97
	traceRouteResultsTable	97
	traceRouteProbeResultsTable	98
	traceRouteHopsTable	99
	Generating Traps	100
	Monitoring Traceroute Test Completion	101
	Gathering Traceroute Test Results	101
	Stopping a Traceroute Test	103
	Interpreting Traceroute Variables	103
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	109
	Example: Configuring Interface Settings for a Routing Instance	110
	Configuring Access Lists for SNMP Access over Routing Instances	111
	Trap Support for Routing Instances	112
	MIB Support Details	112
Chapter 10	Understanding the Junos OS MIB Support	123
	Standard SNMP MIBs Supported on Devices Running Junos OS	123
	Juniper Networks Enterprise-Specific MIBs	123
	Loading MIB Files to a Network Management System	123
Chapter 11	Summary of SNMP Configuration Statements	127
	access-list	127
	agent-address	128
	authorization	128
	categories	129
	client-list	129

client-list-name	130
clients	130
commit-delay	131
community	132
contact	133
description	133
destination-port	134
engine-id	134
enterprise-oid	134
filter-duplicates	135
filter-interfaces	135
interface	136
location	136
logical-system	137
logical-system-trap-filter	138
name	138
nonvolatile	139
oid	139
routing-instance	140
routing-instance-access	141
snmp	141
source-address	142
targets	142
traceoptions	143
trap-group	145
trap-options	146
version	147
view	148
view (Associating a MIB View with a Community)	148
view (Configuring a MIB View)	149
Chapter 12	Summary of SNMPv3 Configuration Statements 151
address	151
address-mask	152
authentication-md5	152
authentication-none	153
authentication-password	154
authentication-sha	155
community-name	156
engine-id	157
group	158
group (Configuring Group Name)	158
group (Defining Access Privileges for an SNMPv3 Group)	159
retry-count	159
timeout	160
local-engine	161
message-processing-model	162

notify	163
notify-filter	164
notify-filter (Applying to the Management Target)	164
notify-filter (Configuring the Profile Name)	164
notify-view	165
oid	165
parameters	166
port	166
privacy-3des	167
privacy-aes128	168
privacy-des	169
privacy-none	169
privacy-password	170
read-view	171
remote-engine	172
routing-instance	173
security-level	174
security-level (Defining Access Privileges)	174
security-level (Generating SNMP Notifications)	175
security-model	176
security-model (Access Privileges)	176
security-model (Group)	177
security-model (SNMP Notifications)	177
security-name	178
security-name (Community String)	178
security-name (Security Group)	179
security-name (SNMP Notifications)	180
security-to-group	181
snmp-community	181
tag	182
tag-list	182
target-address	183
target-parameters	184
type	185
user	185
usm	186
v3	188
vacm	190
view	190
write-view	191

Part 4

Chapter 13

RMON Alarms and Events

Configuring RMON Alarms and Events	195
Understanding RMON Alarms and Events Configuration	195
Minimum RMON Alarm and Event Entry Configuration	196
Configuring an Alarm Entry and Its Attributes	196
Configuring the Alarm Entry	197
Configuring the Description	197

	Configuring the Falling Event Index or Rising Event Index	197
	Configuring the Falling Threshold or Rising Threshold	198
	Configuring the Interval	198
	Configuring the Falling Threshold Interval	198
	Configuring the Request Type	199
	Configuring the Sample Type	199
	Configuring the Startup Alarm	199
	Configuring the System Log Tag	200
	Configuring the Variable	200
	Configuring an Event Entry and Its Attributes	200
	Example: Configuring an RMON Alarm and Event Entry	201
Chapter 14	Monitoring RMON Alarms and Events	203
	Understanding RMON Alarms	203
	alarmTable	204
	jnxRmonAlarmTable	204
	Using alarmTable to Monitor MIB Objects	205
	Creating an Alarm Entry	205
	Configuring the Alarm MIB Objects	205
	alarmInterval	206
	alarmVariable	206
	alarmSampleType	206
	alarmValue	206
	alarmStartupAlarm	206
	alarmRisingThreshold	206
	alarmFallingThreshold	207
	alarmOwner	207
	alarmRisingEventIndex	207
	alarmFallingEventIndex	207
	Activating a New Row in alarmTable	207
	Modifying an Active Row in alarmTable	207
	Deactivating a Row in alarmTable	208
	Understanding RMON Events	208
	eventTable	208
	Using eventTable to Log Alarms	209
	Creating an Event Entry	209
	Configuring the MIB Objects	209
	eventType	209
	eventCommunity	210
	eventOwner	210
	eventDescription	210
	Activating a New Row in eventTable	211
	Deactivating a Row in eventTable	211
Chapter 15	Summary of RMON Alarm and Event Configuration Statements	213
	alarm	213
	community	214
	description	214
	event	215
	falling-event-index	215

	falling-threshold	216
	falling-threshold-interval	217
	interval	217
	request-type	218
	rising-event-index	219
	rising-threshold	219
	rmon	220
	sample-type	220
	startup-alarm	221
	syslog-subtag	221
	type	222
	variable	222
Part 5	Health Monitoring	
Chapter 16	Configuring Health Monitoring	225
	Configuring Health Monitoring on Devices Running Junos OS	225
	Monitored Objects	226
	Minimum Health Monitoring Configuration	227
	Configuring the Falling Threshold or Rising Threshold	227
	Configuring the Interval	227
	Log Entries and Traps	228
	Example: Configuring Health Monitoring	228
Chapter 17	Summary of Health Monitoring Configuration Statements	229
	falling-threshold	229
	health-monitor	230
	interval	230
	rising-threshold	231
Part 6	Monitoring Service Quality	
Chapter 18	Monitoring Service Quality in Service Provider Networks	235
	Understanding Measurement Points, Key Performance Indicators, and Baseline Values	235
	Values	235
	Measurement Points	235
	Basic Key Performance Indicators	236
	Setting Baselines	237
	Understanding RMON for Monitoring Service Quality	237
	Setting Thresholds	237
	RMON Command-Line Interface	238
	RMON Event Table	239
	RMON Alarm Table	239
	Troubleshooting RMON	240
	Defining and Measuring Network Availability	241
	Defining Network Availability	241
	Monitoring the SLA and the Required Bandwidth	243
	Measuring Availability	243
	Real-Time Performance Monitoring	244
	Measuring Health	246

	Measuring Performance	253
	Measuring Class of Service	255
	Inbound Firewall Filter Counters per Class	256
	Monitoring Output Bytes per Queue	257
	Dropped Traffic	258
Part 7	Accounting Options	
Chapter 19	Accounting Options Overview	263
	Accounting Options Overview	263
Chapter 20	Configuring Accounting Options	265
	Accounting Options Configuration	265
	Accounting Options—Full Configuration	265
	Minimum Accounting Options Configuration	266
	Configuring Accounting-Data Log Files	268
	Configuring the Storage Location of the File	269
	Configuring the Maximum Size of the File	269
	Configuring the Maximum Number of Files	270
	Configuring the Start Time for File Transfer	270
	Configuring the Transfer Interval of the File	270
	Configuring Archive Sites	271
	Configuring the Interface Profile	271
	Configuring Fields	272
	Configuring the File Information	272
	Configuring the Interval	272
	Example: Configuring the Interface Profile	272
	Configuring the Filter Profile	274
	Configuring the Counters	274
	Configuring the File Information	274
	Configuring the Interval	275
	Example: Configuring a Filter Profile	275
	Example: Configuring Interface-Specific Firewall Counters and Filter Profiles . .	276
	Understanding Source Class Usage and Destination Class Usage Options . . .	278
	Configuring SCU or DCU	278
	Creating Prefix Route Filters in a Policy Statement	279
	Applying the Policy to the Forwarding Table	279
	Enabling Accounting on Inbound and Outbound Interfaces	279
	Configuring SCU on a Virtual Loopback Tunnel Interface	281
	Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC	281
	Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface	281
	Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface	282
	Configuring Class Usage Profiles	282
	Configuring a Class Usage Profile	282
	Configuring the File Information	283
	Configuring the Interval	283
	Creating a Class Usage Profile to Collect Source Class Usage Statistics . . .	283

	Creating a Class Usage Profile to Collect Destination Class Usage Statistics	284
	Configuring the MIB Profile	284
	Configuring the File Information	285
	Configuring the Interval	285
	Configuring the MIB Operation	285
	Configuring MIB Object Names	286
	Example: Configuring a MIB Profile	286
	Configuring the Routing Engine Profile	286
	Configuring Fields	287
	Configuring the File Information	287
	Configuring the Interval	287
	Example: Configuring a Routing Engine Profile	287
Chapter 21	Summary of Accounting Options Configuration Statements	289
	accounting-options	289
	archive-sites	290
	class-usage-profile	291
	counters	292
	destination-classes	292
	fields	293
	fields (for Interface Profiles)	293
	fields (for Routing Engine Profiles)	294
	file	295
	file (Associating with a Profile)	295
	file (Configuring a Log File)	296
	files	296
	filter-profile	297
	interface-profile	298
	interval	299
	mib-profile	300
	nonpersistent	301
	object-names	301
	operation	302
	routing-engine-profile	302
	size	303
	source-classes	303
	start-time	304
	transfer-interval	304
Part 8	Index	
	Index	307
	Index of Statements and Commands	313

List of Figures

Part 3	Simple Network Management Protocol (SNMP)	
Chapter 7	Configuring SNMPv3	49
	Figure 1: Inform Request and Response	75
Chapter 9	SNMP Support for Routing Instances	105
	Figure 2: SNMP Data for Routing Instances	106
Part 6	Monitoring Service Quality	
Chapter 18	Monitoring Service Quality in Service Provider Networks	235
	Figure 3: Network Entry Points	236
	Figure 4: Setting Thresholds	238
	Figure 5: Regional Points of Presence	241
	Figure 6: Measurements to Each Router	242
	Figure 7: Network Behavior During Congestion	256

List of Tables

	About This Guide	xxiii
	Table 1: Notice Icons	xxvii
	Table 2: Text and Syntax Conventions	xxvii
Part 1	Network Management Introduction	
Chapter 1	Network Management Overview	3
	Table 3: Device Management Features in Junos OS	4
Part 3	Simple Network Management Protocol (SNMP)	
Chapter 5	Configuring SNMP	23
	Table 4: SNMP Tracing Flags	44
Chapter 8	SNMP Remote Operations	85
	Table 5: Results in pingProbeHistoryTable: After the First Ping Test	93
	Table 6: Results in pingProbeHistoryTable: After the First Probe of the Second Test	94
	Table 7: Results in pingProbeHistoryTable: After the Second Ping Test	94
	Table 8: traceRouteProbeHistoryTable	102
Chapter 9	SNMP Support for Routing Instances	105
	Table 9: MIB Support for Routing Instances (Juniper Networks MIBs)	112
	Table 10: Class 1 MIB Objects (Standard and Juniper MIBs)	116
	Table 11: Class 2 MIB Objects (Standard and Juniper MIBs)	120
	Table 12: Class 3 MIB Objects (Standard and Juniper MIBs)	121
	Table 13: Class 4 MIB Objects (Standard and Juniper MIBs)	122
Part 5	Health Monitoring	
Chapter 16	Configuring Health Monitoring	225
	Table 14: Monitored Object Instances	226
Part 6	Monitoring Service Quality	
Chapter 18	Monitoring Service Quality in Service Provider Networks	235
	Table 15: RMON Event Table	239
	Table 16: RMON Alarm Table	239
	Table 17: jnxRmon Alarm Extensions	240
	Table 18: Real-Time Performance Monitoring Configuration Options	244
	Table 19: Health Metrics	246
	Table 20: Counter Values for vlan-ccc Encapsulation	252

Table 21: Performance Metrics	253
Table 22: Inbound Traffic Per Class	257
Table 23: Inbound Counters	257
Table 24: Outbound Counters for ATM Interfaces	258
Table 25: Outbound Counters for Non-ATM Interfaces	258
Table 26: Dropped Traffic Counters	258

Part 7

Chapter 19

Accounting Options

Accounting Options Overview	263
Table 27: Types of Accounting Profiles	263

About This Guide

This preface provides the following guidelines for using the *Junos[®] OS Network Management Configuration Guide*:

- Junos OS Documentation and Release Notes on page xxiii
- Objectives on page xxiv
- Audience on page xxiv
- Supported Platforms on page xxiv
- Using the Indexes on page xxv
- Using the Examples in This Manual on page xxv
- Documentation Conventions on page xxvi
- Documentation Feedback on page xxviii
- Requesting Technical Support on page xxviii

Junos OS Documentation and Release Notes

For a list of related Junos OS documentation, see <http://www.juniper.net/techpubs/software/junos/>.

If the information in the latest release notes differs from the information in the documentation, follow the *Junos OS Release Notes*.

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

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at <http://www.juniper.net/books>.

Objectives

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



NOTE: For additional information about the Junos OS—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 router or switch.

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 Platforms

For the features described in this manual, the Junos OS currently supports the following 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.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
```

```
        family inet {  
            address 10.0.0.1/24;  
        }  
    }  
}
```

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see the [Junos OS CLI User Guide](#).

Documentation Conventions

Table 1 on page xxvii 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 xxvii 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: <code>user@host> configure</code>
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	<code>stub <default-metric metric>;</code>

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

Convention	Description	Examples
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast <i>(string1 string2 string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop address; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to 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 or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or 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/us/en/local/pdf/resource-guides/7100059-en.pdf> .
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/> .
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- 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: <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 Device Management Functions in Junos OS on page 3

Understanding Device Management Functions in Junos OS

After you have installed a 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 OS network management features work in conjunction with an operations support system (OSS) to manage the devices within the network. Junos OS can assist you in performing these management tasks, as described in Table 3 on page 4.

Table 3: Device Management Features in Junos OS

Task	Junos OS Feature
Fault management	<p>Monitor and see faults using:</p> <ul style="list-style-type: none"> Operational mode commands—For more information about operational mode commands, see the Junos OS System Basics and Services Command Reference, Junos OS Interfaces Command Reference, and Junos OS Routing Protocols and Policies Command Reference. SNMP MIBs—For more information about SNMP MIBs supported by Junos OS, see “Standard SNMP MIBs Supported by Junos OS” and “Juniper Networks Enterprise-Specific MIBs” in the Junos OS SNMP MIBs and Traps Reference. Standard SNMP traps—For more information about standard SNMP traps, see the “Standard SNMP Traps Supported on Devices Running Junos OS” in the Junos OS SNMP MIBs and Traps Reference. Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see “Juniper Networks Enterprise-Specific SNMP Traps” in the Junos OS SNMP MIBs and Traps Reference. System log messages—For more information about how to configure system log messages, see the Junos OS System Basics Configuration Guide. For more information about how to view system log messages, see the Junos OS System Log Messages Reference.
Configuration management	<ul style="list-style-type: none"> Configure router attributes using the command-line interface (CLI), the Junos XML management protocol, and the NETCONF XML management protocol. For more information about configuring the router using the CLI, see the Junos OS System Basics Configuration Guide. For more information about configuring the router using the APIs, see the Junos XML Management Protocol Guide and NETCONF XML Management Protocol Guide. Configuration Management MIB—For more information about the Configuration Management MIB, see the “Configuration Management MIB” in the Junos OS SNMP MIBs and Traps Reference.

Table 3: Device Management Features in Junos OS (*continued*)

Task	Junos OS Feature
Accounting management	<p>Perform the following accounting-related tasks:</p> <ul style="list-style-type: none"> Collect statistics for interfaces, firewall filters, destination classes, source classes, and the Routing Engine. For more information about collecting statistics, see “Accounting Options Configuration” on page 265. Use interface-specific traffic statistics and other counters, available in the Standard Interfaces MIB, Juniper Networks enterprise-specific extensions to the Interfaces MIB, and media-specific MIBs, such as the enterprise-specific ATM MIB. Use per-ATM virtual circuit (VC) counters, available in the enterprise-specific ATM MIB. For more information about the ATM MIB, see the Junos OS SNMP MIBs and Traps Reference. 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 about classes, see “Destination Class Usage MIB” and “Source Class Usage MIB” in the Junos OS SNMP MIBs and Traps Reference, “Configuring Class Usage Profiles” on page 282, the Junos OS Network Interfaces Configuration Guide, and the Junos OS Routing Policy Configuration Guide. Count packets as part of a firewall filter. For more information about firewall filter policies, see “Juniper Networks Enterprise-Specific MIBs” in the Junos OS SNMP MIBs and Traps Reference and the Junos OS Routing Policy Configuration Guide. Sample traffic, collect the samples, and send the collection to a host running the CAIDA cflowd utility. For more information about CAIDA and cflowd, see the Junos OS Routing Policy Configuration Guide.
Performance management	<p>Monitor performance in the following ways:</p> <ul style="list-style-type: none"> Use operational mode commands. For more information about monitoring performance using operational mode commands, see the Junos OS System Basics and Services Command Reference. Use firewall filter. For more information about performance monitoring using firewall filters, see the Junos OS Routing Policy Configuration Guide. Sample traffic, collect the samples, and send the samples to a host running the CAIDA cflowd utility. For more information about CAIDA and cflowd, see the Junos OS Routing Policy Configuration Guide. Use the enterprise-specific Class-of-Service MIB. For more information about this MIB, see the “Class-of-Service MIB” in the Junos OS SNMP MIBs and Traps Reference.
Security management	<p>Assure security in your network in the following ways:</p> <ul style="list-style-type: none"> Control access to the router and authenticate users. For more information about access control and user authentication, see the Junos OS System Basics Configuration Guide. Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see “Configuring the Local Engine ID” on page 53 and “Tracing SNMP Activity on a Device Running Junos OS” on page 42.

**Related
Documentation**

- [Understanding the Integrated Local Management Interface on page 15](#)
- [Understanding the SNMP Implementation in Junos OS on page 19](#)
- [Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235](#)
- [Accounting Options Overview on page 263](#)

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 Junos OS, 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 OS Configuration Statements and Commands](#).

```
[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;
    start-time 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;
    object-names {
        mib-object-name;
    }
    operation operation-name;
}
routing-engine-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

- Related Documentation**
- Accounting Options Overview on page 263
 - Accounting Options Configuration on page 265

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 Junos OS, 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 OS Configuration Statements and Commands](#).

```
[edit]
snmp {
    client-list client-list-name {
        ip-addresses;
    }
    community community-name {
        authorization authorization;
        client-list-name client-list-name;
        clients {
            address <restrict>;
        }
    }
}
```

```

logical-system logical-system-name {
    routing-instance routing-instance-name;
    clients {
        address <restrict>;
    }
}
routing-instance routing-instance-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
        regular-expression>;
    flag flag;
}
trap-group group-name {
    categories {
        category;
    }
    destination-port port-number;
}

```

```

routing-instance instance;
logical-system logical-system-name;
targets {
    address;
}
version (all | v1 | v2);
}
trap-options {
    agent-address outgoing-interface;
    source-address address;
    enterprise-oid;
    logical-system logical-system-name {
        routing-instance routing-instance-name {
            source-address address;
        }
    }
    routing-instance routing-instance-name {
        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;
        logical-system logical-system;
        port port-number;
        retry-count number;
        routing-instance instance;
        tag-list tag-list;
        target-parameters target-parameters-name;
        timeout seconds;
    }
    target-parameters target-parameters-name {
        notify-filter profile-name;
        parameters {
            message-processing-model (v1 | v2c | v3);
            security-level (authentication | none | privacy);
            security-model (usm | v1 | v2c);
            security-name security-name;
        }
    }
}
usm {
    local-engine {
        user username {

```



```

    authentication-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 | context-prefix context-prefix){
                security-model (any | usm | v1 | v2c) {
                    security-level (authentication | none | privacy) {
                        notify-view view-name;
                        read-view view-name;
                        write-view view-name;
                    }
                }
            }
        }
    }
    security-to-group {
        security-model (usm | v1 | v2c) {
            security-name security-name {
                group group-name;
            }
        }
    }
}
}
view view-name {
    oid object-identifier (include | exclude);
}
}

```

**Related
Documentation**

- Understanding the SNMP Implementation in Junos OS on page 19
- Configuring SNMP on a Device Running Junos OS on page 24

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 the Integrated Local Management Interface on page 15

Understanding the 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 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).

Junos OS supports only two ILMI 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 OS Network Interfaces Configuration Guide](#). For information about displaying ILMI statistics, see the [Junos OS Interfaces Command Reference](#). For more information about the ILMI MIB, see the ATM Forum at <http://www.atmforum.com/>.

Related Documentation

- Understanding Device Management Functions in Junos OS on page 3

PART 3

Simple Network Management Protocol (SNMP)

- [SNMP Overview on page 19](#)
- [Configuring SNMP on page 23](#)
- [SNMPv3 Overview on page 47](#)
- [Configuring SNMPv3 on page 49](#)
- [SNMP Remote Operations on page 85](#)
- [SNMP Support for Routing Instances on page 105](#)
- [Understanding the Junos OS MIB Support on page 123](#)
- [Summary of SNMP Configuration Statements on page 127](#)
- [Summary of SNMPv3 Configuration Statements on page 151](#)

CHAPTER 4

SNMP Overview

This chapter contains the following topic:

- Understanding the SNMP Implementation in Junos OS on page 19

Understanding the SNMP Implementation in Junos OS

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 OS.

This topic includes the following sections:

- SNMP Architecture on page 19
- Junos OS 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 MIB, the collection of objects that can be viewed or changed by the SNMP manager.

The SNMP manager collects information about 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:

- SNMP MIBs on page 20
- SNMP Traps and Informs on page 20

SNMP MIBs

A MIB 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 website, www.ietf.org, and compile them into your NMS, if necessary.

For a list of standard supported MIBs, see Standard SNMP MIBs Supported by Junos OS in the *Junos OS SNMP MIBs and Traps Reference*.

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 in the *Junos OS SNMP MIBs and Traps Reference*.

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 website, www.ietf.org.

For more information about standard traps supported by the Junos OS, see Standard SNMP Traps Supported on Devices Running Junos OS in the *Junos OS SNMP MIBs and Traps Reference*.

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 about enterprise-specific traps supported by the Junos OS, see Juniper Networks Enterprise-Specific SNMP Traps in the *Junos OS SNMP MIBs and Traps*

Reference. For information about 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 about SNMP informs, see “Configuring SNMP Informs” on page 75.

SNMP Trap Queuing

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

The Junos OS 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 OS checks for availability of routes every 30 seconds and sends the traps from the destination queue in a round-robin fashion. If the trap delivery fails, the trap is added back to the queue, and the delivery attempt counter and the next delivery attempt timer for the queue are reset. Subsequent attempts occur at progressive intervals of 1 minute, 2 minutes, 4 minutes, and 8 minutes. The maximum delay between the attempts is 8 minutes, and the maximum number of attempts is 10. After 10 unsuccessful attempts, the destination queue and all the traps in the queue are deleted.

The Junos OS also has a throttle mechanism to control the number of traps (**throttle threshold**; default value of 500 traps) sent during a particular time period (**throttle interval**; default of 5 seconds) and to ensure consistency in trap traffic, especially when a 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 trap queues (that is, the throttle queue and the destination queue combined) is 40,000 traps. However, on EX Series switches, the maximum size of the trap queue is 1000 traps. The maximum size of any one queue is 20,000 traps for devices other than EX Series switches. On EX Series switches, the maximum size of one queue is 500 traps. If a trap is sent from a destination queue when the throttle queue has exceeded the maximum size, the trap is added back to the 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 OS 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 OS System Basics Configuration Guide*.

For more information about system logging severity levels for standard traps, see Standard SNMP Version 1 Traps and Standard SNMP Version 2 Traps in the [Junos OS SNMP MIBs and Traps Reference](#). For more information about system logging severity levels for enterprise-specific traps, see Juniper Networks Enterprise-Specific SNMP Version 1 Traps and Juniper Networks Enterprise-Specific SNMP Version 2 Traps in the [Junos OS SNMP MIBs and Traps Reference](#).

Junos OS SNMP Agent Features

The Junos OS 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 OS 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 OS SNMP agent software accepts IPv4 and IPv6 addresses for transport over IPv4 and IPv6. For IPv6, the Junos OS supports the following features:

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

Related Documentation

- [SNMPv3 Overview on page 47](#)
- [Configuring SNMP on a Device Running Junos OS on page 24](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 8](#)

CHAPTER 5

Configuring SNMP

This chapter contains the following topics:

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the System Contact on a Device Running Junos OS on page 26
- Configuring the System Location for a Device Running Junos OS on page 26
- Configuring the System Description on a Device Running Junos OS on page 27
- Filtering Duplicate SNMP Requests on page 27
- Configuring the Commit Delay Timer on page 28
- Configuring the System Name on page 28
- Configuring the SNMP Community String on page 29
- Examples: Configuring the SNMP Community String on page 30
- Adding a Group of Clients to an SNMP Community on page 30
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 32
- Configuring SNMP Trap Options on page 33
- Configuring SNMP Trap Groups on page 36
- Example: Configuring SNMP Trap Groups on page 38
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 39
- Example: Configuring Secured Access List Checking on page 39
- Filtering Interface Information Out of SNMP Get and GetNext Output on page 40
- Configuring MIB Views on page 40
- Example: Ping Proxy MIB on page 41
- Tracing SNMP Activity on a Device Running Junos OS on page 42
- Example: Tracing SNMP Activity on page 45
- Configuring the Local Engine ID on page 46

Configuring SNMP on a Device Running Junos OS

By default, SNMP is disabled on devices running Junos OS. To enable SNMP on a router or switch, 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;
        falling-threshold-interval seconds;
        interval seconds;
        request-type (get-next-request | get-request | walk-request);
        rising-event-index index;
        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
        regular-expression>;
    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);
}
}

```

Related Documentation

- Understanding the SNMP Implementation in Junos OS on page 19
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Complete SNMPv3 Configuration Statements on page 50

Configuring the System Contact on a Device Running Junos OS

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 (" ").

To define a system contact name that contains spaces:

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

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the System Location for a Device Running Junos OS on page 26
- Configuring the System Description on a Device Running Junos OS on page 27
- Configuring the System Name on page 28
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

Configuring the System Location for a Device Running Junos OS

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 (" ").

To specify the system location:

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

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the System Contact on a Device Running Junos OS on page 26
- Configuring the System Description on a Device Running Junos OS on page 27
- Configuring the System Name on page 28
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

Configuring the System Description on a Device Running Junos OS

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 (" ").

To specify the system description:

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

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the System Contact on a Device Running Junos OS on page 26
- Configuring the System Location for a Device Running Junos OS on page 26
- Configuring the System Name on page 28
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

Filtering Duplicate SNMP Requests

By default, filtering duplicate **get**, **getNext**, and **getBulk** SNMP requests is disabled on devices running Junos OS. If a network management station retransmits a **Get**, **GetNext**, or **GetBulk** SNMP request too frequently to the router, that request 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. Junos OS 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;
```

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 39
- Filtering Interface Information Out of SNMP Get and GetNext Output on page 40
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

Configuring the Commit Delay Timer

When a router or switch first receives an SNMP nonvolatile **Set** request, a Junos OS XML protocol 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 Junos OS XML protocol 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 OS CLI User Guide](#).

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

Configuring the System Name

Junos OS 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 (" ").

To specify the system name override:

```
[edit]
snmp {
  name "snmp1";
}
```

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuring the System Contact on a Device Running Junos OS on page 26
- Configuring the System Location for a Device Running Junos OS on page 26
- Configuring the System Description on a Device Running Junos OS on page 27

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

Configuring the SNMP Community String

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

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

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

The default authorization level for a community is **read-only**. To allow **Set** requests within a community, you need to define that community as **authorization read-write**. For **Set** requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the **view** statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information about the **view** statement, see "Configuring MIB Views" on page 40.

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.

Related Documentation

- Adding a Group of Clients to an SNMP Community on page 30
- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Examples: Configuring the SNMP Community String on page 30

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.
    }
  }
}
```

Related Documentation

- [Configuring the SNMP Community String on page 29](#)

Adding a Group of Clients to an SNMP Community

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

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

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

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

The following example shows how to define a client list:

```
[edit]
snmp {
  client-list clientlist1 {
    10.1.1.1/32;
    10.2.2.2/32;
  }
}
```

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

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

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

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

- Related Documentation**
- [client-list](#)
 - [client-list-name](#)

Configuring SNMP Trap Options and Groups on a Device Running Junos OS

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 device running Junos OS 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);
}
```

- Related Documentation**
- [Configuring SNMP Trap Options on page 33](#)
 - [Configuring SNMP Trap Groups on page 36](#)
 - [Configuring SNMP on a Device Running Junos OS on page 24](#)
 - [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 8](#)

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 about 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;
  enterprise-oid
  logical-system
  routing-instance
  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 36.

This topic contains the following sections:

- Configuring the Source Address for SNMP Traps on page 33
- Configuring the Agent Address for SNMP Traps on page 35
- Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps on page 36

Configuring the Source Address for SNMP Traps

You can configure the source address of trap packets in many ways: **lo0**, a valid IPv4 address configured on one of the router interfaces, a logical-system address, or the address of a routing-instance. The value **lo0** indicates that the source address of the SNMP trap packets is set to the lowest loopback address configured on the interface **lo0**.

You can configure the source address of trap packets in one of the following formats:

- a valid IPv4 address configured on one of the router interfaces
- **lo0**; that is the lowest loopback address configured on the interface **lo0**.
- a logical-system name
- a routing-instance name

A valid IPv4 Address As the Source Address 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.

The Lowest Loopback Address As the Source Address To specify the source address of the SNMP traps so that they use the lowest loopback address configured on the interface **lo0** as the source address, 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;
    }
  }
}
```

To configure the loopback address as the source address of trap packets:

```
[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.

Logical System Name as the Source Address To specify a logical system name as the source address of SNMP traps, include the **logical-system** *logical-system-name* statement at the **[edit snmp trap-options]** hierarchy level.

For example, the following configuration sets logical system name **ls1** as the source address of SNMP traps:

```
[edit snmp]
  trap-options{
    logical-system ls1;
  }
```

Routing Instance Name as the Source Address To specify a routing instance name as the source address of SNMP traps, include the **routing-instance** *routing-instance-name* statement at the **[edit snmp trap-options]** hierarchy level.

For example, the following configuration sets the routing instance name **ri1** as the source address for SNMP traps:

```
[edit snmp]
  trap-options {
    routing-instance ri1;
  }
```

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;
  }
```

To 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.

Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps

The **snmpTrapEnterprise** object helps you identify the enterprise that has defined the trap. Typically, the **snmpTrapEnterprise** object appears as the last varbind in enterprise-specific SNMP version 2 traps. However, starting Release 10.0, Junos OS enables you to add the **snmpTrapEnterprise** object identifier to standard SNMP traps as well.

To add **snmpTrapEnterprise** to standard traps, include the **enterprise-oid** statement at the **[edit snmp trap-options]** hierarchy level. If the **enterprise-oid** statement is not included in the configuration, **snmpTrapEnterprise** is added only for enterprise-specific traps.

```
[edit snmp]
trap-options {
  enterprise-oid;
}
```

Related Documentation

- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 32
- Configuring SNMP Trap Groups on page 36
- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8

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 the category to which the traps belong, see the Standard SNMP Traps Supported on Devices Running Junos OS and Juniper Networks Enterprise-Specific SNMP Traps topics in the *Junos OS SNMP MIBs and Traps Reference*.

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
- **vrp-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 about the **version** statement, see **version**.

Related Documentation

- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 32
- Configuring SNMP Trap Options on page 33
- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Example: Configuring SNMP Trap Groups on page 38

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;
  }
}

```

**Related
Documentation**

- Configuring SNMP Trap Groups on page 36
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 32
- Configuring SNMP Trap Options on page 33

Configuring the Interfaces on Which SNMP Requests Can Be Accepted

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

```

[edit snmp]
interface [ interface-names ];

```

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

**Related
Documentation**

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Example: Configuring Secured Access List Checking on page 39
- Configuring SNMP

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 ];
}

```

- Related Documentation**
- [Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 39](#)
 - [Filtering Interface Information Out of SNMP Get and GetNext Output on page 40](#)
 - [Configuring SNMP on a Device Running Junos OS on page 24](#)
 - [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 8](#)

Filtering Interface Information Out of SNMP Get and GetNext Output

Junos OS enables you to filter out information related to specific interfaces from the output of SNMP **Get** and **GetNext** requests performed on interface-related MIBs such as IF MIB, ATM MIB, RMON MIB, and the Juniper Networks enterprise-specific IF MIB.

You can use the following options of the **filter-interfaces** statement at the **[edit snmp]** hierarchy level to specify the interfaces that you want to exclude from SNMP **Get** and **GetNext** queries:

- **interfaces**—Interfaces that match the specified regular expressions.
- **all-internal-interfaces**—Internal interfaces.

```
[edit]
snmp {
  filter-interfaces {
    interfaces {
      interface1;
      interface2;
    }
    all-internal-interfaces;
  }
}
```

However, note that these settings are limited to SNMP operations, and the users can continue to access information related to the interfaces (including those hidden using the **filter-interfaces** options) using the appropriate Junos OS command-line interface (CLI) commands.

- Related Documentation**
- [Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 39](#)
 - [Configuring SNMP on a Device Running Junos OS on page 24](#)
 - [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 8](#)

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



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

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

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

For more information about the Ping MIB, see RFC 2925 and the PING MIB topic in the [Junos OS SNMP MIBs and Traps Reference](#).

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Example: Ping Proxy MIB on page 41
- view (Configuring a MIB View)
- view (Associating MIB View with a Community)
- oid

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;
}
```

The following configuration prevents the *no-ping-mib* community from accessing Ping MIB and *jnxPingMIB* objects. However, this configuration does not prevent the *no-ping-mib* community from accessing any other MIB object that is supported on the device.

```
[edit snmp]
view no-ping-mib-view {
  oid 1.3.6.1.2.1.80 exclude; # deny access to pingMIB objects
  oid jnxPingMIB exclude; # deny access to jnxPingMIB objects
}
community no-ping-mib {
  authorization read-write;
  view ping-mib-view;
}
```

**Related
Documentation**

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Configuring MIB Views on page 40
- **view (Configuring a MIB View)** on page 149
- **oid** on page 139

Tracing SNMP Activity on a Device Running Junos OS

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

By default, Junos OS does not trace any SNMP activity. If you include the **traceoptions** statement at the **[edit snmp]** hierarchy level, the default tracing behavior is:

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

- Log files can be accessed only by the user who configured the tracing operation.

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

```
[edit snmp]
traceoptions {
  file <files number> <match regular-expression> <size size> <world-readable |
    no-world-readable>;
  flag flag;
  no-remote-trace;
}
```

These statements are described in the following sections:

- Configuring the Number and Size of SNMP Log Files on page 43
- Configuring Access to the Log File on page 43
- Configuring a Regular Expression for Lines to Be Logged on page 44
- Configuring the Trace Operations on page 44

Configuring the Number and Size of SNMP Log Files

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

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

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

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

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

Configuring Access to the Log File

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

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

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

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

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

Configuring a Regular Expression for Lines to Be Logged

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

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

```
[edit snmp traceoptions]
file filename match regular-expression;
```

Configuring the Trace Operations

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

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

Table 4 on page 44 describes the meaning of the SNMP tracing flags.

Table 4: SNMP Tracing Flags

Flag	Description	Default Setting
all	Log all operations.	Off
configuration	Log reading of the configuration at the [edit snmp] hierarchy level.	Off
database	Log events involving storage and retrieval in the events database.	Off
events	Log important events.	Off

Table 4: SNMP Tracing Flags (*continued*)

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

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

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

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

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 24
- Configuration Statements at the [edit snmp] Hierarchy Level on page 8
- Example: Tracing SNMP Activity on page 45
- Configuring SNMP

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;  
    }  
}
```

**Related
Documentation**

- [Configuring SNMP on a Device Running Junos OS on page 24](#)
- [Tracing SNMP Activity on a Device Running Junos OS on page 42](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 8](#)

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 53.

CHAPTER 6

SNMPv3 Overview

This chapter contains the following topic:

- SNMPv3 Overview on page 47

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 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 object identifiers (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 54](#)
- [Configuring MIB Views on page 40](#)
- [Defining Access Privileges for an SNMP Group on page 58](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 65](#)
- [Configuring SNMP Informs on page 75](#)

**Related
Documentation**

- [Complete SNMPv3 Configuration Statements on page 50](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

CHAPTER 7

Configuring SNMPv3

This chapter contains the following topics:

- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Configuring the Local Engine ID on page 53
- Creating SNMPv3 Users on page 54
- Configuring the SNMPv3 Authentication Type on page 55
- Configuring the Encryption Type on page 56
- Defining Access Privileges for an SNMP Group on page 58
- Configuring the Access Privileges Granted to a Group on page 59
- Example: Access Privilege Configuration on page 62
- Assigning Security Model and Security Name to a Group on page 63
- Example: Security Group Configuration on page 64
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the SNMPv3 Trap Notification on page 66
- Example: Configuring SNMPv3 Trap Notification on page 67
- Configuring the Trap Notification Filter on page 67
- Configuring the Trap Target Address on page 68
- Example: Configuring the Tag List on page 71
- Defining and Configuring the Trap Target Parameters on page 72
- Configuring SNMP Informs on page 75
- Configuring the Remote Engine and Remote User on page 76
- Example: Configuring the Remote Engine ID and Remote Users on page 77
- Configuring the Inform Notification Type and Target Address on page 77
- Example: Configuring the Inform Notification Type and Target Address on page 79
- Configuring the SNMPv3 Community on page 79
- Example: SNMPv3 Community Configuration on page 81
- 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;
  logical-system logical-system;
  port port-number;
  retry-count number;
  routing-instance instance;
  tag-list tag-list;
  target-parameters target-parameters-name;
  timeout seconds;
}
target-parameters target-parameters-name {
  notify-filter profile-name;
  parameters {
    message-processing-model (v1 | v2c | v3);
    security-level (authentication | none | privacy);
    security-model (usm | v1 | v2c);
    security-name security-name;
  }
}
usm {
  (local-engine | 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 | context-prefix context-prefix){
                security-model (any | usm | v1 | v2c) {
                    security-level (authentication | none | privacy) {
                        notify-view view-name;
                        read-view view-name;
                        write-view view-name;
                    }
                }
            }
        }
    }
}
security-to-group {
    security-model (usm | v1 | v2c) {
        security-name security-name {
            group group-name;
        }
    }
}
}
}

```

Related Documentation

- [Creating SNMPv3 Users on page 54](#)
- [Configuring MIB Views on page 40](#)
- [Defining Access Privileges for an SNMP Group on page 58](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 65](#)
- [Configuring SNMP Informs on page 75](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

Minimum SNMPv3 Configuration on a Device Running Junos OS

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



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

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

```

    }
  }
}
security-to-group {
  security-model (usm | v1 | v2c) {
    security-name security-name {
      group group-name;
    }
  }
}
}
}

```

Related Documentation

- Creating SNMPv3 Users on page 54
- Configuring MIB Views on page 40
- Defining Access Privileges for an SNMP Group on page 58
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring SNMP Informs on page 75
- Complete SNMPv3 Configuration Statements on page 50
- Example: SNMPv3 Configuration on page 82

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 router.

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 are based on the previous engine ID. For the engine ID, we recommend using the master IP address of the device if the device has multiple routing engines and has the master IP address configured. Alternatively, you can use the MAC address of fxp0 if the device has only one Routing Engine.

Related Documentation

- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: SNMPv3 Configuration on page 82

Creating SNMPv3 Users

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



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

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

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

username is the name that identifies the SNMPv3 user.

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

```
[edit snmp v3 usm local-engine user username]
authentication-md5 {
    authentication-password authentication-password;
}
authentication-sha {
    authentication-password authentication-password;
}
authentication-none;
privacy-aes128 {
    privacy-password privacy-password;
}
privacy-des {
    privacy-password privacy-password;
}
privacy-3des {
```

```

    privacy-password privacy-password;
  }
  privacy-none;

```

Related Documentation

- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: Creating SNMPv3 Users Configuration
- Example: SNMPv3 Configuration on page 82

Configuring the SNMPv3 Authentication Type

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

This topic includes the following sections:

- Configuring MD5 Authentication on page 55
- Configuring SHA Authentication on page 55
- Configuring No Authentication on page 56

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 router or switch:

- The password must be at least eight characters long.
- The password can include alphabetic, numeric, and special characters, but it cannot include 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 router or switch:

- The password must be at least eight characters long.
- The password can include alphabetic, numeric, and special characters, but it cannot include 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;
```

Related Documentation

- Configuring the Encryption Type on page 56
- Defining Access Privileges for an SNMP Group on page 58
- Configuring the Access Privileges Granted to a Group on page 59
- Assigning Security Model and Security Name to a Group on page 63
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Configuring the Encryption Type

By default, encryption is set to none.



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

Before you configure the **privacy-des**, **privacy-3des** and **privacy-aes128** statements, you must install the **jcrypto** package, and either restart the SNMP process or reboot the router.

This topic includes the following sections:

- Configuring the Advanced Encryption Standard Algorithm on page 56
- Configuring the Data Encryption Algorithm on page 57
- Configuring Triple DES on page 57
- Configuring No Encryption on page 58

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:

```
[edit 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 router or switch:

- The password must be at least eight characters long.
- The password can include alphabetic, numeric, and special characters, but it cannot include 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 router or switch:

- The password must be at least eight characters long.
- The password can include alphabetic, numeric, and special characters, but it cannot include 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:

```

[edit 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 router or switch:

- The password must be at least eight characters long.
- The password can include alphabetic, numeric, and special characters, but it cannot include 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;
```

Related Documentation

- Configuring the SNMPv3 Authentication Type on page 55
- Defining Access Privileges for an SNMP Group on page 58
- Configuring the Access Privileges Granted to a Group on page 59
- Assigning Security Model and Security Name to a Group on page 63
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Defining Access Privileges for an SNMP Group

The SNMP 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 40.

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, read (**get**, **getNext**, or **getBulk**) write (**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 | context-prefix context-prefix) {
      security-model (any | usm | v1 | v2c) {
        security-level (authentication | none | privacy) {
          notify-view view-name;
          read-view view-name;
          write-view view-name;
        }
      }
    }
  }
}
security-to-group {
  security-model (usm | v1 | v2c) {
    security-name security-name {
      group group-name;
    }
  }
}
```

Related Documentation

- Configuring the SNMPv3 Authentication Type on page 55
- Configuring the Access Privileges Granted to a Group on page 59
- Assigning Security Model and Security Name to a Group on page 63
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Configuring the Access Privileges Granted to a Group

This topic includes the following sections:

- Configuring the Group on page 59
- Configuring the Security Model on page 60
- Configuring the Security Level on page 60
- Associating MIB Views with an SNMP User Group on page 60

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 | context-prefix *context-prefix*)]** hierarchy level:

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

```

notify-view view-name;
read-view view-name;
write-view view-name;

```



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 | context-prefix *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 40.

This section describes the following topics related to this configuration:

- Configuring the Notify View on page 61
- Configuring the Read View on page 61
- Configuring the Write View on page 62

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 | context-prefix *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 | context-prefix
  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 | context-prefix *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 | context-prefix
  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 | context-prefix *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 | context-prefix
  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.

Related Documentation

- Configuring the SNMPv3 Authentication Type on page 55
- Defining Access Privileges for an SNMP Group on page 58
- Assigning Security Model and Security Name to a Group on page 63
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: Access Privilege Configuration on page 62

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;
        }
      }
    }
  }
  context-prefix lr1/ri1 { # routing instance ri1 in logical system lr1
    security-model usm {
      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;
            }
        }
    }
}
}

```

Related Documentation

- [Configuring the Access Privileges Granted to a Group on page 59](#)
- [Complete SNMPv3 Configuration Statements on page 50](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

Assigning Security Model and Security Name to a Group

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 63](#)
- [Assigning Security Names to Groups on page 64](#)
- [Configuring the Group on page 64](#)

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

Assigning Security Names to Groups

To associate a security name with an SNMPv3 user, or a v1 or v2 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;
```

For SNMPv3, the **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 54. For information about configuring a community string, see “Configuring the SNMPv3 Community” on page 79.



NOTE: The USM security name is separate from the SNMPv1 and SNMPv2c security name. If you support SNMPv1 and SNMPv2c in addition to SNMPv3, 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 SNMPv3 users, or 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 58.

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;
  }
}

```

Related Documentation

- Assigning Security Model and Security Name to a Group on page 63
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Configuring SNMPv3 Traps on a Device Running Junos OS

In SNMPv3, you create traps and informs by configuring the **notify**, **target-address**, and **target-parameters** parameters. Traps are unconfirmed notifications, whereas informs are confirmed notifications. This section describes how to configure SNMP traps. For information about configuring SNMP informs, see “Configuring SNMP Informs” on page 75.

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



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

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

```

[edit snmp v3]
notify name {
  tag tag-name;
  type trap;
}
notify-filter name {
  oid object-identifier (include | exclude);
}
target-address target-address-name {
  address address;
  address-mask address-mask;
  logical-system logical-system;
}

```

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

**Related
Documentation**

- Configuring the SNMPv3 Trap Notification on page 66
- Configuring the Trap Notification Filter on page 67
- Configuring the Trap Target Address on page 68
- Defining and Configuring the Trap Target Parameters on page 72
- Configuring SNMP Informs on page 75
- Configuring the Remote Engine and Remote User on page 76
- Configuring the Inform Notification Type and Target Address on page 77
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

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, Junos OS 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 to which this notification is sent. This notification is sent to all the target-addresses that have this tag in their tag list. The ***tag-name*** is not included in the notification.

trap is the type of notification.



NOTE: Each notify entry name must be unique.

Junos OS supports two types of notification: **trap** and **inform**.

For information about how to configure the tag list, see “Configuring the Trap Target Address” on page 70.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the Trap Notification Filter on page 67
- Configuring the Trap Target Address on page 68
- Defining and Configuring the Trap Target Parameters on page 72
- Configuring SNMP Informs on page 75
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: Configuring SNMPv3 Trap Notification on page 67

Example: Configuring SNMPv3 Trap Notification

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;
}
```

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Configuring the Trap Notification Filter

SNMPv3 uses the notify filter to define which traps (or which objects from which traps) are 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**). You can also use the wildcard character asterisk (*) in the object identifier (OID) to specify object identifiers that match a particular pattern.

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.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the SNMPv3 Trap Notification on page 66
- Configuring the Trap Target Address on page 68
- Defining and Configuring the Trap Target Parameters on page 72
- Configuring SNMP Informs on page 75
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

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, Junos OS 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 SNMPv2cc 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;
logical-system logical-system;
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 69
- Configuring the Address Mask on page 69
- Configuring the Port on page 70
- Configuring the Routing Instance on page 70
- Configuring the Trap Target Address on page 70
- Applying Target Parameters on page 71

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 79.

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 Trap Target Address

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 an example of tag list configuration, see “Example: Configuring the Tag List” on page 71.

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 66.



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.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the SNMPv3 Trap Notification on page 66
- Configuring the Trap Notification Filter on page 67
- Defining and Configuring the Trap Target Parameters on page 72
- Configuring SNMP Informs on page 75
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: Configuring the Tag List on page 71

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, Junos OS 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;
}
```

Related Documentation

- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 65](#)
- [Configuring the Trap Target Address on page 68](#)
- [Complete SNMPv3 Configuration Statements on page 50](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

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 73](#)
- [Configuring the Target Parameters on page 73](#)

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 67.

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

This section includes the following topics:

- Configuring the Message Processing Model on page 73
- Configuring the Security Model on page 73
- Configuring the Security Level on page 74
- Configuring the Security Name on page 74

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.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the SNMPv3 Trap Notification on page 66
- Configuring the Trap Notification Filter on page 67
- Configuring the Trap Target Address on page 68
- Configuring SNMP Informs on page 75

- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

Configuring SNMP Informs

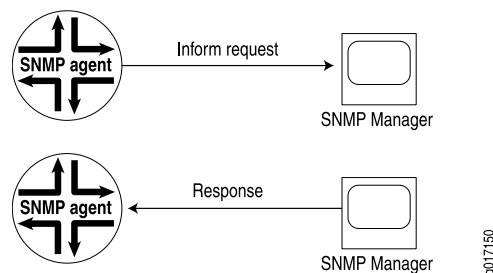
Junos OS supports two types of notifications: traps and informs. With traps, the receiver does not send any acknowledgment when it receives a trap. Therefore, the sender cannot determine if the trap was received. A trap may be lost because a problem occurred during transmission. To increase reliability,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

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

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

Informs are more reliable than traps, but they consume more network, router, and switch resources (see Figure 1 on page 75). Unlike a trap, an inform is held in memory until a response is received or the timeout is reached. Also, traps are sent only once, whereas an inform may be retried several times. Use informs when it is important that the SNMP manager receive all notifications. However, if you are more concerned about network traffic, or router and switch memory, use traps.

Figure 1: Inform Request and Response



For information about configuring SNMP traps, see “Configuring SNMPv3 Traps on a Device Running Junos OS” on page 65.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring the Remote Engine and Remote User on page 76
- Configuring the Inform Notification Type and Target Address on page 77
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

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.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring SNMP Informs on page 75
- Configuring the Inform Notification Type and Target Address on page 77
- Complete SNMPv3 Configuration Statements on page 50

- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: Configuring the Remote Engine ID and Remote Users on page 77

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
0BEhSreW87s24aUjsY4ZDjq.RhcyWLNdbg4Zs
YJDHkTQ69Apu1EcyrvWQF/tuOREYg4ajHmPQF39
Ygz3n6At8XxNYgik.PTz7-ikmfn6vW8XVw";
      }
    }
    privacy-des {
      privacy-key "$9$MZZXxdwYgJUjIKJGiH5T69Au0IrlM7NbeK24
aJDjO1IRylM8Xbwg1R24aJDjHqm5n/Ap0ORhn6evLXbwmf5T
/CRhSyKM5QEcleW87-Vbs4JGD.mT-VwgaZkqfTznAphSrlM8yr
Wx7dsYTzF36Atu0IEcpuNdwYoa69CuRhcycleM8rlaZGjq.O1IEhr";
    }
  }
}
```

Related Documentation

- Configuring the Remote Engine and Remote User on page 76
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

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;
  logical-system logical-system;
  port port-number;
  retry-count number;
```

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

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 Trap Target Address” on page 70.

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.

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.

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.

**Related
Documentation**

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 65
- Configuring SNMP Informs on page 75

- [Configuring the Remote Engine and Remote User on page 76](#)
- [Complete SNMPv3 Configuration Statements on page 50](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)
- [Example: Configuring the Inform Notification Type and Target Address on page 79](#)

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;
  retry-count 3;
  tag-list tl1;
  address-mask 255.255.255.0;
  target-parameters tp1;
  timeout 30;
}
target-parameters tp1 {
  parameters {
    message-processing-model v3;
    security-model usm;
    security-level privacy;
    security-name u10;
  }
  notify-filter nf1;
}
```

Related Documentation

- [Configuring the Inform Notification Type and Target Address on page 77](#)
- [Complete SNMPv3 Configuration Statements on page 50](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

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 80
- Configuring the Security Names on page 81
- Configuring the Tag on page 81

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 command-line interface (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.

Related Documentation

- Creating SNMPv3 Users on page 54
- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52
- Example: SNMPv3 Community Configuration on page 81

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
  }
}
```

- Related Documentation**
- [Configuring the SNMPv3 Community on page 79](#)
 - [Complete SNMPv3 Configuration Statements on page 50](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52](#)

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;
    }
  }
}
}
```

Related Documentation

- Complete SNMPv3 Configuration Statements on page 50
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

CHAPTER 8

SNMP Remote Operations

This chapter contains the following topics:

- SNMP Remote Operations Overview on page 85
- Using the Ping MIB for Remote Monitoring Devices Running Junos OS on page 88
- Starting a Ping Test on page 88
- Monitoring a Running Ping Test on page 90
- Gathering Ping Test Results on page 93
- Stopping a Ping Test on page 94
- Interpreting Ping Variables on page 94
- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
- Starting a Traceroute Test on page 96
- Monitoring a Running Traceroute Test on page 97
- Monitoring Traceroute Test Completion on page 101
- Gathering Traceroute Test Results on page 101
- Stopping a Traceroute Test on page 103
- Interpreting Traceroute Variables on page 103

SNMP Remote Operations Overview

A SNMP remote operation is any process on the router that can be controlled remotely using SNMP. Junos OS currently provides support for two SNMP remote operations: the Ping 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

Junos OS 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 the PING MIB and Traceroute MIB topics in the *Junos OS SNMP MIBs and Traps Reference*.

This topic covers the following sections:

- SNMP Remote Operation Requirements on page 86
- Setting SNMP Views on page 86
- Setting Trap Notification for Remote Operations on page 87
- Using Variable-Length String Indexes on page 87
- Enabling Logging on page 88

SNMP Remote Operation Requirements

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

Setting SNMP Views

All remote operation MIBs supported by Junos OS require that the SNMP clients have read-write privileges. The default SNMP configuration of Junos OS 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:

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

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 about the **community** statement, see “Configuring the SNMP Community String” on page 29 and **community**.

For more information about the **view** statement, see “Configuring MIB Views” on page 40, **view (Associating a MIB View with a Community)**, and **view (Configuring a MIB View)**.

Setting Trap Notification for Remote Operations

In addition to configuring the remote operations MIB for trap notification, you must also configure Junos OS. 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 about trap groups, see “Configuring SNMP Trap Groups” on page 36.

Using Variable-Length String Indexes

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

Junos OS 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 object identifier (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 object identifier (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 about 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 about 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:

```
[edit]
snmp {
  traceoptions {
    flag general;
  }
}
```

For more information about traceoptions, see “Tracing SNMP Activity on a Device Running Junos OS” on page 42.

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).

- Related Documentation**
- Using the Ping MIB for Remote Monitoring Devices Running Junos OS on page 88
 - Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95

Using the Ping MIB for Remote Monitoring Devices Running Junos OS

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.

- Related Documentation**
- SNMP Remote Operations Overview on page 85
 - Starting a Ping Test on page 88
 - Monitoring a Running Ping Test on page 90
 - Gathering Ping Test Results on page 93
 - Stopping a Ping Test on page 94
 - Interpreting Ping Variables on page 94

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 object identifier (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 86.

There are two ways to start a ping test:

- Using Multiple Set Protocol Data Units (PDUs) on page 89
- Using a Single Set PDU on page 89

Using Multiple Set Protocol Data Units (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**

Junos OS 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 90
- **pingProbeHistoryTable** on page 91
- **Generating Traps** on page 92

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 about **pingProbeHistoryTable**, see “pingProbeHistoryTable” on page 91.

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

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



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

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

pingProbeHistoryTable

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

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

The maximum number of **pingProbeHistoryTable** entries created for a given test is limited by **pingCtlMaxRows**. If **pingCtlMaxRows** is set to 0, no **pingProbeHistoryTable** entries are 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 are 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 36 and “Example: Setting Trap Notification for Remote Operations” on page 87.

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 about **pingResultsOperStatus**, see “pingResultsTable” on page 90. For more information about Ping MIB traps, see “Generating Traps” on page 92.

The statistics calculated and then stored in **pingResultsTable** include:

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

You can also consult **pingProbeHistoryTable** for more detailed information about 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 5 on page 93.

Table 5: 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 6 on page 94.

Table 6: 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 7 on page 94.

Table 7: 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 for Remote Monitoring Devices Running Junos OS

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.

Related Documentation

- SNMP Remote Operations Overview on page 85
- Starting a Traceroute Test on page 96
- Monitoring a Running Traceroute Test on page 97
- Monitoring Traceroute Test Completion on page 101
- Gathering Traceroute Test Results on page 101
- Stopping a Traceroute Test on page 103
- Interpreting Traceroute Variables on page 103

Starting a Traceroute Test

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

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

For all other values, defaults are chosen unless otherwise specified.

traceRouteCtlOwnerIndex and **traceRouteCtlTestName** are used as the index, so their values are specified as part of the OID. To create a row, set **traceRouteCtlRowStatus** to **createAndWait** or **createAndGo** on a row that does not already exist. A value of **active** for **traceRouteCtlRowStatus** indicates that all necessary information has been specified and the test can begin; **traceRouteCtlAdminStatus** can be set to **enabled**. An SNMP **Set** request that sets **traceRouteCtlRowStatus** to **active** will fail if the necessary information in the row is not specified or is inconsistent. For information about how to configure a view, see “Setting SNMP Views” on page 86.

There are two ways to start a traceroute test:

- Using Multiple Set PDUs on page 96
- Using a Single Set PDU on page 96

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 OS 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**

- Related Documentation**
- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
 - Monitoring a Running Traceroute Test on page 97
 - SNMP Remote Operations Overview on page 85
 - Monitoring Traceroute Test Completion on page 101
 - Gathering Traceroute Test Results on page 101
 - Stopping a Traceroute Test on page 103
 - Interpreting Traceroute Variables on page 103

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 97
- **traceRouteProbeResultsTable** on page 98
- **traceRouteHopsTable** on page 99
- Generating Traps on page 100

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 is **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 time-to-live (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 time-to-live or 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 36 and “Example: Setting Trap Notification for Remote Operations” on page 87.

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.

Related Documentation

- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
- Monitoring a Running Traceroute Test on page 97
- SNMP Remote Operations Overview on page 85
- Starting a Traceroute Test on page 96
- Gathering Traceroute Test Results on page 101
- Stopping a Traceroute Test on page 103
- Interpreting Traceroute Variables on page 103

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 about **traceResultsOperStatus**, see “traceRouteResultsTable” on page 97. For more information about Traceroute MIB traps, see the Generating Traps section in “Monitoring a Running Traceroute Test” on page 97.

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 about 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 8 on page 102.

Table 8: traceRouteProbeHistoryTable

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

Table 8: traceRouteProbeHistoryTable (*continued*)

HistoryIndex	HistoryHopIndex	HistoryProbeIndex
38	8	3
39	8	4
40	8	5

Related Documentation

- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
- Monitoring a Running Traceroute Test on page 97
- SNMP Remote Operations Overview on page 85
- Starting a Traceroute Test on page 96
- Monitoring Traceroute Test Completion on page 101
- Stopping a Traceroute Test on page 103
- Interpreting Traceroute Variables on page 103

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**.

Related Documentation

- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
- Monitoring a Running Traceroute Test on page 97
- SNMP Remote Operations Overview on page 85
- Starting a Traceroute Test on page 96
- Monitoring Traceroute Test Completion on page 101
- Gathering Traceroute Test Results on page 101
- Interpreting Traceroute Variables on page 103

Interpreting Traceroute Variables

This topic contains information about 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.

**Related
Documentation**

- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS on page 95
- Monitoring a Running Traceroute Test on page 97
- SNMP Remote Operations Overview on page 85
- Starting a Traceroute Test on page 96
- Monitoring Traceroute Test Completion on page 101
- Gathering Traceroute Test Results on page 101
- Stopping a Traceroute Test on page 103

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 109
- Example: Configuring Interface Settings for a Routing Instance on page 110
- Configuring Access Lists for SNMP Access over Routing Instances on page 111
- Trap Support for Routing Instances on page 112
- MIB Support Details on page 112

Understanding SNMP Support for Routing Instances

Junos OS enables SNMP managers for all routing instances to request and manage SNMP data related to the corresponding routing instances and logical system networks.

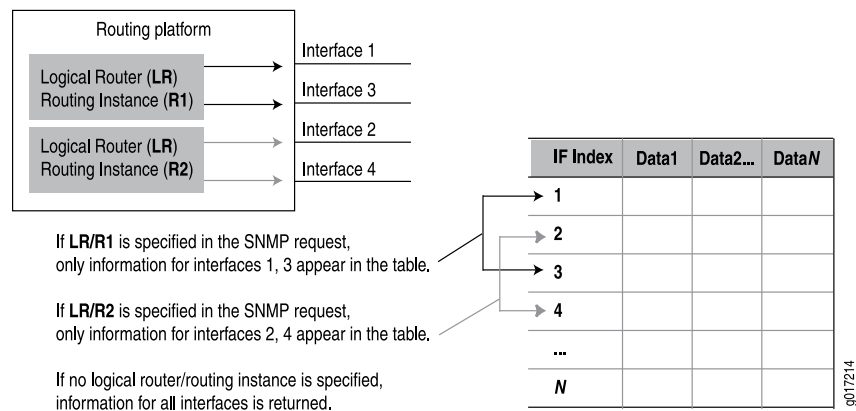
In Junos OS:

- 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 OS 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 nonrouting table objects, all instances are exposed.
- For routing table objects, only those associated with the default routing instance are 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.

Related Documentation

- Support Classes for MIB Objects on page 106
- Trap Support for Routing Instances on page 112
- 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 109
- Configuring Access Lists for SNMP Access over Routing Instances on page 111

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 112 for a list of the objects associated with each class.

**Related
Documentation**

- Understanding SNMP Support for Routing Instances on page 105
- Trap Support for Routing Instances on page 112

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 protocol data unit (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 virtual LAN (VLAN) spanning-tree instance (VSTP on MX Series 3D Universal Edge Routers), 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.

**Related
Documentation**

- Understanding SNMP Support for Routing Instances on page 105
- Enabling SNMP Access over Routing Instances on page 108
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109

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 at the **[edit snmp]** hierarchy level:

```
[edit snmp]
  routing-instance-access;
```

If this statement is not included in the SNMP configuration, SNMP managers from routing instances other than the default routing instance cannot access SNMP information.

**Related
Documentation**

- Understanding SNMP Support for Routing Instances on page 105
- Identifying a Routing Instance on page 107
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109
- Configuring Access Lists for SNMP Access over Routing Instances on page 111

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;
      }
    }
  }
}
```

Related Documentation

- Understanding SNMP Support for Routing Instances on page 105
- Identifying a Routing Instance on page 107
- Enabling SNMP Access over Routing Instances on page 108
- Configuring Access Lists for SNMP Access over Routing Instances on page 111
- Example: Configuring Interface Settings for a Routing Instance on page 110

Example: Configuring Interface Settings for a Routing Instance

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

```
[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]
INFrttd {
  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 **INFrttd** with the SNMP community **public**:

```
router# snmpwalk -Os router1 INFrttd@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

```

- Related Documentation**
- Understanding SNMP Support for Routing Instances on page 105
 - Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109

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.

- Related Documentation**
- Understanding SNMP Support for Routing Instances on page 105
 - Enabling SNMP Access over Routing Instances on page 108
 - Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109

Trap Support for Routing Instances

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.

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).

- Related Documentation**
- Understanding SNMP Support for Routing Instances on page 105
 - Support Classes for MIB Objects on page 106
 - MIB Support Details on page 112

MIB Support Details

Table 9 on page 112 shows enterprise-specific MIB objects supported by Junos OS 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 9: MIB Support for Routing Instances (Juniper Networks MIBs)

Object	Support Class	Description/Notes
jnxProducts(1)	–	Product Object IDs
jnxServices(2)	–	Services
jnxMibs(3)	Class 3	Objects are exposed only for the default logical system.
jnxBoxAnatomy(1)		

Table 9: MIB Support for Routing Instances (Juniper Networks MIBs) (*continued*)

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

Table 9: 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 are 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 are exposed.
jnxRpf(17) jnxRpfStatsTable(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance are exposed.
jnxCfgMgmt(18)	Class 3	Objects are 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 are exposed.
jnxSonet(20) jnxSonetAlarmTable(1)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance are 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 are 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 are exposed.
apsMIB(24)	Class 3	Objects are exposed only for the default logical system.
jnxChassisDefines(25)	Class 3	Objects are exposed only for the default logical system.

Table 9: MIB Support for Routing Instances (Juniper Networks MIBs) (continued)

Object	Support Class	Description/Notes
jnxVpnMIB(26)	Class 2	All instances within a logical system are 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 are exposed.
jnxCollectorMIB(28)	Class 1	Only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance are exposed.
jnxHistory(29)	—	—
jnxSpMIB(32)	Class 3	Objects are exposed only for the default logical system.

Table 10 on page 116 shows Class 1 MIB objects (standard and enterprise-specific MIBs) supported by Junos OS. With Class 1 objects, only those logical interfaces (and their parent physical interfaces) that belong to a specific routing instance are exposed.

Table 10: Class 1 MIB Objects (Standard and Juniper MIBs)

Class	MIB	Objects
Class 1	802.3ad.mib	(dot3adAgg) MIB objects: dot3adAggTable dot3adAggPortListTable (dot3adAggPort) 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	sonetMediumTable (and related MIB objects)

Table 10: Class 1 MIB Objects (Standard and Juniper MIBs) (*continued*)

Class	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 10: 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	vrrp 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 10: Class 1 MIB Objects (Standard and Juniper MIBs) (*continued*)

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

Table 11 on page 120 shows Class 2 MIB objects (standard and enterprise-specific MIBs) supported by Junos OS. With Class 2 objects, all instances within a logical system are exposed. Data will not be segregated down to the routing instance level.

Table 11: 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-bgp.mib	jnxBgpM2Experiment
	jnx-bgp-mib2.mib	jnxBgpM2Experiment

Table 12 on page 121 shows Class 3 MIB objects (standard and enterprise-specific MIBs) supported by Junos OS. With Class 3, objects are exposed only for the default logical system.

Table 12: 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 13 on page 122 shows Class 4 MIB objects (standard and enterprise-specific MIBs) supported by Junos OS. With Class 4 objects, data is not segregated by routing instance. All instances are exposed.

Table 13: 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	sysAppLOBJ
	jnx-firewall.mib	jnxFirewalls
	jnx-ipv6.mib	jnxIpv6

**Related
Documentation**

- Understanding SNMP Support for Routing Instances on page 105
- Support Classes for MIB Objects on page 106
- Trap Support for Routing Instances on page 112

CHAPTER 10

Understanding the Junos OS MIB Support

This chapter contains the following topics:

- Standard SNMP MIBs Supported on Devices Running Junos OS on page 123
- Juniper Networks Enterprise-Specific MIBs on page 123
- Loading MIB Files to a Network Management System on page 123

Standard SNMP MIBs Supported on Devices Running Junos OS

For information about the Standard SNMP MIBs supported on devices running Junos OS, see the Standard SNMP MIBs Supported by Junos OS topic in the *Junos OS SNMP MIBs and Traps Reference*.

Juniper Networks Enterprise-Specific MIBs

For information about the Juniper Networks Enterprise-Specific MIBs supported on devices running Junos OS, see the Juniper Networks Enterprise-Specific MIBs topic in the *Junos OS SNMP MIBs and Traps Reference*.

Loading MIB Files to a Network Management System

For your network management system (NMS) to identify and understand the MIB objects used by the Junos OS, you must first load the MIB files to your NMS using a MIB compiler. A MIB compiler is a utility that parses the MIB information such as the MIB object name, IDs, and data type for the NMS.

You can download the Junos MIB package from the **Enterprise-Specific MIBs and Traps** section of the Junos OS Technical Publications index page at <http://www.juniper.net/techpubs/software/junos/index.html>. The Junos MIB package is available in .zip and .tar packages. You can download the appropriate format based on your requirements.

The Junos MIB package contains two folders: **StandardMibs** and **JuniperMibs**. The **StandardMibs** folder contains the standard MIBs and RFCs that are supported on devices running the Junos OS, whereas the **JuniperMibs** folder contains the Juniper Networks enterprise-specific MIBs.

To load MIB files that are required for managing and monitoring devices running the Junos OS:

1. Go to the Junos OS Technical Publications index page (<http://www.juniper.net/techpubs/software/junos/index.html>).
2. Click the tab that corresponds to the Junos OS Release for which you want to download the MIB files.
3. On the selected tab, click the + (plus) sign that corresponds to the **Enterprise-Specific MIBs and Traps** section to expand the section.
4. Click the **TAR** or **ZIP** link that corresponds to the **Enterprise MIBs** link under the **Enterprise-Specific MIBs and Traps** section to download the Junos MIB package.
5. Decompress the file (.tar or .zip) using an appropriate utility.
6. Load the standard MIB files (from the **StandardMibs** folder) in the following order:



NOTE: Some of the MIB compilers that are commonly used have the standard MIBs preloaded on them. If the standard MIBs are already loaded on the MIB compiler that you are using, skip this step and proceed to Step 7.

- a. **mib-SNMPv2-SMI.txt**
 - b. **mib-SNMPv2-TC.txt**
 - c. **mib-IANAifType-MIB.txt**
 - d. **mib-IANA-RTPROTO-MIB.txt**
 - e. **mib-rfc1907.txt**
 - f. **mib-rfc2011a.txt**
 - g. **mib-rfc2012a.txt**
 - h. **mib-rfc2013a.txt**
 - i. **mib-rfc2863a.txt**
7. Load the remaining standard MIB files.



NOTE: You must follow the order specified in this procedure, and ensure that all standard MIBs are loaded before you load the enterprise-specific MIBs. There might be dependencies that require a particular MIB to be present on the compiler before loading some other MIB. You can find such dependencies listed in the **IMPORT** section of the MIB file.

8. Load the Juniper Networks enterprise-specific SMI MIB, **mib-jnx-smi.txt**, and the following optional SMI MIBs based on your requirements:

- **mib-jnx-js-smi.txt**—(Optional) For Juniper Security MIB tree objects
- **mib-jnx-ex-smi.txt**—(Optional) For EX Series Ethernet Switches
- **mib-jnx-exp.txt**—(Recommended) For Juniper Networks experimental MIB objects

9. Load the remaining enterprise-specific MIBs from the **JuniperMibs** folder.



TIP: While loading a MIB file, if the compiler returns an error message saying that any of the objects is undefined, open the MIB file using a text editor and ensure that all the MIB files listed in the **IMPORT** section are loaded on the compiler. If any of the MIB files listed in the **IMPORT** section is not loaded on the compiler, load that MIB file, and then try to load the MIB file that failed to load.

For example, the enterprise-specific PING MIB, **mib-jnx-ping.txt**, has dependencies on RFC 2925, DiSMAN-PING-MIB, **mib-rfc2925a.txt**. If you try to load **mib-jnx-ping.txt** before loading **mib-rfc2925a.txt**, the compiler returns an error message saying that certain objects in **mib-jnx-ping.txt** are undefined. Load **mib-rfc2925a.txt**, and then try to load **mib-jnx-ping.txt**. The enterprise-specific PING MIB, **mib-jnx-ping.txt**, then loads without any issue.

Related Documentation

- Standard SNMP MIBs Supported by Junos OS
- Juniper Networks Enterprise-Specific MIBs

CHAPTER 11

Summary of SNMP Configuration Statements

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

access-list

Syntax [edit snmp]
 routing-instance-access {
 access-list {
 routing-instance;
 routing-instance restrict;
 }
 }

Hierarchy Level [edit snmp routing-instance-access]

Release Information Statement introduced in Junos OS Release 8.4.

Description Create access lists to control SNMP agents in routing instances from accessing SNMP information. To enable the SNMP agent on a routing instance to access SNMP information, specify the routing instance name. To disable the SNMP agent on a routing instance from accessing SNMP information, include the routing-instance name followed by the **restrict** keyword.

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

Related Documentation • [routing-instance-access on page 141](#)

agent-address

Syntax	<code>agent-address outgoing-interface;</code>
Hierarchy Level	[edit snmp trap-options]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Set the agent address of all SNMPv1 traps generated by this router or switch. Currently, the only option is outgoing-interface , which sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.
Options	outgoing-interface —Value of the agent address of all SNMPv1 traps generated by this router or switch. The outgoing-interface option sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap. Default: disabled (the agent address is not specified in SNMPv1 traps).
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Agent Address for SNMP Traps on page 35

authorization

Syntax	<code>authorization <i>authorization</i>;</code>
Hierarchy Level	[edit snmp community <i>community-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Set the access authorization for SNMP Get , GetBulk , GetNext , and Set requests.
Options	authorization —Access authorization level: <ul style="list-style-type: none">read-only—Enable Get, GetNext, and GetBulk requests.read-write—Enable all requests, including Set requests. You must configure a view to enable Set requests. Default: read-only
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the SNMP Community String on page 29

categories

Syntax	<code>categories { category; }</code>
Hierarchy Level	<code>[edit snmp trap-group group-name]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define the types of traps that are sent to the targets of the named trap group.
Default	If you omit the categories statement, all trap types are included in trap notifications.
Options	category —Name of a trap type: authentication , chassis , configuration , link , remote-operations , rmon-alarm , routing , sonet-alarms , startup , or vrp-events .
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring SNMP Trap Groups on page 36

client-list

Syntax	<code>client-list client-list-name { ip-addresses; }</code>
Hierarchy Level	<code>[edit snmp]</code>
Release Information	Statement introduced in Junos OS Release 8.5. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define a list of SNMP clients.
Options	client-list-name —Name of the client list. ip-addresses —IP addresses of the SNMP clients to be added to the client list,
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Adding a Group of Clients to an SNMP Community on page 30

client-list-name

Syntax	<code>client-list-name <i>client-list-name</i>;</code>
Hierarchy Level	<code>[edit snmp community <i>community-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.5. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Add a client list or prefix list to an SNMP community.
Options	<i>client-list-name</i> —Name of the client list or prefix list.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Adding a Group of Clients to an SNMP Community on page 30

clients

Syntax	<pre>clients { <i>address</i> <restrict>; }</pre>
Hierarchy Level	<code>[edit snmp community <i>community-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the IPv4 or IPv6 addresses of the SNMP client hosts that are authorized to use this community.
Default	If you omit the clients statement, all SNMP clients using this community string are authorized to access the router.
Options	<i>address</i> —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 <i>address</i> options. <i>restrict</i> —(Optional) Do not allow the specified SNMP client to access the router.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the SNMP Community String on page 29

commit-delay

Syntax	commit-delay <i>seconds</i> ;
Hierarchy Level	[edit snmp nonvolatile]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure the timer for the SNMP Set reply and start of the commit.
Options	<i>seconds</i> —Delay between an affirmative SNMP Set reply and start of the commit. Default: 5 seconds
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Commit Delay Timer on page 28

community

Syntax	<pre>community <i>community-name</i> { authorization <i>authorization</i>; client-list-name <i>client-list-name</i>; clients { address restrict; } view <i>view-name</i>; }</pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>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.</p> <p>The SNMP client application specifies an SNMP community name in Get, GetBulk, GetNext, and Set SNMP requests.</p>
Default	If you omit the community statement, all SNMP requests are denied.
Options	<p><i>community-name</i>—Community string. If the name includes spaces, enclose it in quotation marks (" ").</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">Configuring the SNMP Community String on page 29

contact

Syntax	<code>contact <i>contact</i>;</code>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define the value of the MIB II sysContact object, which is the contact person for the managed system.
Options	contact —Name of the contact person. If the name includes spaces, enclose it in quotation marks (" ").
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the System Contact on a Device Running Junos OS on page 26

description

Syntax	<code>description <i>description</i>;</code>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Define the value of the MIB II sysDescription object, which is the description of the system being managed.
Options	description —System description. If the name includes spaces, enclose it in quotation marks (" ").
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the System Description on a Device Running Junos OS on page 27

destination-port

Syntax	<code>destination-port <i>port-number</i>;</code>
Hierarchy Level	[edit snmp trap-group]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Assign a trap port number other than the default.
Default	If you omit this statement, the default port is 162.
Options	<i>port-number</i> —SNMP trap port number.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Groups on page 36

engine-id

See `engine-id`

enterprise-oid

Syntax	<code>enterprise-oid;</code>
Hierarchy Level	[edit snmp trap-options]
Release Information	Statement introduced in Junos OS Release 10.0
Description	Add the snmpTrapEnterprise object, which shows the association between an enterprise-specific trap and the organization that defined the trap, to standard SNMP traps. By default, the snmpTrapEnterprise object is added only to the enterprise-specific traps. When the enterprise-oid statement is included in the configuration, snmpTrapEnterprise is added to all the traps generated from the device.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Options on page 33

filter-duplicates

Syntax	filter-duplicates;
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Filter duplicate Get , GetNext , or GetBulk SNMP requests.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Filtering Duplicate SNMP Requests on page 27

filter-interfaces

Syntax	<pre>filter-interfaces { interfaces { all-internal-interfaces; interface 1; interface 2; } }</pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 9.4. Statement introduced in Junos OS Release 9.4 for EX Series Switches.
Description	Filter out information related to specific interfaces from the output of SNMP Get and GetNext requests performed on interface-related MIBs.
Options	<p>all-internal-interfaces—Filters out information from SNMP Get and GetNext requests for the specified interfaces.</p> <p>interfaces—Specifies the interfaces to filter out from the output of SNMP Get and GetNext requests.</p>
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Filtering Interface Information Out of SNMP Get and GetNext Output on page 40


interface

Syntax	<code>interface [<i>interface-names</i>];</code>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the interfaces on which SNMP requests can be accepted.
Default	If you omit this statement, SNMP requests entering the router or switch through any interface are accepted.
Options	<i>interface-names</i> —Names of one or more logical interfaces.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 39

location

Syntax	<code>location <i>location</i>;</code>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define the value of the MIB II sysLocation object, which is the physical location of the managed system.
Options	<i>location</i> —Location of the local system. You must enclose the name within quotation marks (" ").
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the System Location for a Device Running Junos OS on page 26

logical-system

Syntax	<code>logical-system <i>logical-system-name</i> { routing-instance <i>routing-instance-name</i>; }</code>
Hierarchy Level	<code>[edit snmp community <i>community-name</i>], [edit snmp trap-group], [edit snmp trap-options] [edit snmp v3target-address <i>target-address-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 9.3 Statement introduced in Junos OS Release 9.0 for EX Series switches.
	 <p>NOTE: The <code>logical-system</code> statement replaces the <code>logical-router</code> statement, and is backward-compatible with Junos OS Release 8.3 and later.</p>
Description	<p>Specify a logical system name for SNMP v1 and v2c clients.</p> <p>Include at the <code>[edit snmp trap-options]</code> hierarchy level to specify a logical-system address as the source address of an SNMP trap.</p> <p>Include at the <code>[edit snmp v3 target-address]</code> hierarchy level to specify a logical-system name as the destination address for an SNMPv3 trap or inform.</p>
Options	<p><i>logical-system-name</i>—Name of the logical system.</p> <p><i>routing-instance routing-instance-name</i>—Statement to specify a routing instance associated with the logical system.</p>
Required Privilege Level	<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>
Related Documentation	<ul style="list-style-type: none"> Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109 Configuring the Trap Target Address on page 68

logical-system-trap-filter

Syntax	logical-system-trap-filter;
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 8.4.
Description	Restrict the routing instances from receiving traps that are not related to the logical system networks to which they belong.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Trap Support for Routing Instances on page 112

name

Syntax	name <i>name</i> ;
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Set the system name from the command-line interface.
Options	<i>name</i> —System name override.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the System Name on page 28

nonvolatile

Syntax	nonvolatile { commit-delay <i>seconds</i> ; }
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. The commit-delay statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure options for SNMP Set requests. The statement is explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Commit Delay Timer on page 28 commit-delay on page 131

oid

Syntax	oid <i>object-identifier</i> (exclude include);
Hierarchy Level	[edit snmp view <i>view-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify an object identifier (OID) used to represent a subtree of MIB objects.
Options	<p>exclude—Exclude the subtree of MIB objects represented by the specified OID.</p> <p>include—Include the subtree of MIB objects represented by the specified OID.</p> <p>object-identifier—OID used to represent a subtree of MIB objects. All MIB objects represented by this statement have the specified OID as a prefix. You can specify the OID using either a sequence of dotted integers or a subtree name.</p>
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring MIB Views on page 40

routing-instance

Syntax	<code>routing-instance <i>routing-instance-name</i>;</code>
Hierarchy Level	<code>[edit snmp community <i>community-name</i>],</code> <code>[edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i>],</code> <code>[edit snmp trap-group <i>group</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.3. Added to the <code>[edit snmp community <i>community-name</i>]</code> hierarchy level in Junos OS Release 8.4. Added to the <code>[edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i>]</code> hierarchy level in Junos OS Release 9.1. Statement introduced in Junos OS Release 9.1 for EX Series switches.
Description	<p>Specify a routing instance for SNMPv1 and SNMPv2 trap targets. All targets configured in the trap group use this routing instance.</p> <p>If the routing instance is defined within a logical system, include the logical-system <i>logical-system-name</i> statement at the <code>[edit snmp community <i>community-name</i>]</code> hierarchy level and specify the routing-instance statement under the <code>[edit snmp community <i>community-name</i> logical-system <i>logical system-name</i>]</code> hierarchy level.</p>
Options	<i>routing-instance-name</i> —Name of the routing instance.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Groups on page 36Configuring the Source Address for SNMP Traps on page 33Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 109

routing-instance-access

Syntax	<pre>[edit snmp] routing-instance-access { access-list { <i>routing-instance</i>; <i>routing-instance</i> restrict; } }</pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 8.4.
Description	Enable SNMP managers in routing instances other than the default routing instance to access SNMP information. For information about the access-list option, see access-list .
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Enabling SNMP Access over Routing Instances on page 108

snmp

Syntax	snmp { ... }
Hierarchy Level	[edit]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure SNMP.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring SNMP on a Device Running Junos OS on page 24

source-address

Syntax	<code>source-address <i>address</i>;</code>
Hierarchy Level	<code>[edit snmp trap-options]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Set the source address of every SNMP trap packet sent by this router to a single address regardless of the outgoing interface. If the source address is not specified, the default is to use the address of the outgoing interface as the source address.
Options	<i>address</i> —Source address of SNMP traps. You can configure the source address of trap packets two ways: lo0 or a valid IPv4 address configured on one of the router interfaces. The value lo0 indicates that the source address of all SNMP trap packets is set to the lowest loopback address configured at interface lo0 . Default: Disabled. (The source address is the address of the outgoing interface.)
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Source Address for SNMP Traps on page 33

targets

Syntax	<code>targets { <i>address</i>; }</code>
Hierarchy Level	<code>[edit snmp trap-group <i>group-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure one or more systems to receive SNMP traps.
Options	<i>address</i> —IPv4 or IPv6 address of the system to receive traps. You must specify an address, not a hostname.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Groups on page 36

traceoptions

Syntax	<pre> traceoptions { file <i>filename</i> <files <i>number</i>> <match <i>regular-expression</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i>; no-remote-trace; } </pre>
Hierarchy Level	[edit snmp]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>file <i>filename</i> option added in Junos OS Release 8.1.</p> <p>world-readable no-world-readable option added in Junos OS Release 8.1.</p> <p>match <i>regular-expression</i> option added in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>The output of the tracing operations is placed into log files in the /var/log directory. Each log file is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:</p> <ul style="list-style-type: none"> • chassisd • craftd • ilmids • mib2d • rmopd • serviced • snmpd
Options	<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, mib2d or snmpd). Use this option to specify another name.</p> <p>files <i>number</i>—(Optional) Maximum number of trace files per SNMP subagent. When a trace file (for example, snmpd) reaches its maximum size, it is archived by being renamed to snmpd.0. The previous snmpd.1 is renamed to snmpd.2, and so on. The oldest archived file is deleted.</p> <p>Range: 2 through 1000 files</p> <p>Default: 10 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements:</p> <ul style="list-style-type: none"> • all—Log all SNMP events. • 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 *regular-expression*—(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

world-readable | no-world-readable—(Optional) By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

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

Related Documentation	<ul style="list-style-type: none">• Tracing SNMP Activity on a Device Running Junos OS on page 42
------------------------------	---

trap-group

Syntax	<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>
Hierarchy Level	[edit snmp]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	Create a named group of hosts to receive the specified trap notifications. The name of the trap group is embedded in SNMP trap notification packets as one variable binding (varbind) known as the community name. At least one trap group must be configured for SNMP traps to be sent.
Options	<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>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring SNMP Trap Groups on page 36

trap-options

Syntax	<pre>trap-options { agent-address outgoing-interface; source-address address; }</pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router or switch to a single address, regardless of the outgoing interface. In addition, you can set the agent address of each SNMPv1 trap. For more information about the contents of SNMPv1 traps, see RFC 1157.</p> <p>The remaining statements are explained separately.</p>
Default	Disabled
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Options on page 33

version

Syntax	version (all v1 v2);
Hierarchy Level	[edit snmp trap-group <i>group-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the version number of SNMP traps.
Default	all—Send an SNMPv1 and SNMPv2 trap for every trap condition.
Options	all—Send an SNMPv1 and SNMPv2 trap for every trap condition. v1—Send SNMPv1 traps only. v2—Send SNMPv2 traps only.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring SNMP Trap Groups on page 36

view


See the following sections:

- [view \(Associating a MIB View with a Community\)](#) on page 148
- [view \(Configuring a MIB View\)](#) on page 149

view (Associating a MIB View with a Community)

Syntax	<code>view view-name;</code>
Hierarchy Level	<code>[edit snmp community community-name]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
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.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the SNMP Community String on page 29

view (Configuring a MIB View)

Syntax	<pre>view view-name { oid object-identifier (include exclude); }</pre>
Hierarchy Level	[edit snmp]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>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.</p>
	<div>  <p>NOTE: To remove an OID completely, use the delete view all oid oid-number command but omit the include parameter.</p> </div>
Options	<p>view-name—Name of the view.</p> <p>The remaining statement is explained separately.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring MIB Views on page 40 Associating MIB Views with an SNMP User Group on page 60 community on page 132

CHAPTER 12

Summary of SNMPv3 Configuration Statements

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

address

Syntax	<code>address <i>address</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the SNMP target address.
Options	<i>address</i> —IPv4 address of the system to receive traps or informs. You must specify an address, not a hostname.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Address on page 69

address-mask

Syntax	<code>address-mask <i>address-mask</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Verify the source addresses for a group of target addresses.
Options	<i>address-mask</i> combined with the address defines a range of addresses.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Address Mask on page 69

authentication-md5

Syntax	<code>authentication-md5 { authentication-password <i>authentication-password</i>; }</code>
Hierarchy Level	<code>[edit snmp v3 usm local-engine user <i>username</i>],</code> <code>[edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure MD5 as the authentication type for the SNMPv3 user.



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

The remaining statement is explained separately.

Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring MD5 Authentication on page 55

authentication-none

Syntax	authentication-none;
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure that there should be no authentication for the SNMPv3 user.



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

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

Related Documentation	<ul style="list-style-type: none"> Configuring No Authentication on page 56
------------------------------	--

authentication-password

Syntax	<code>authentication-password <i>authentication-password</i>;</code>
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i> authentication-md5], [edit snmp v3 usm local-engine user <i>username</i> authentication-sha], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> authentication-md5], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> authentication-sha]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the password for user authentication.
Options	<p><i>authentication-password</i>—Password that a user enters. The password is then converted into a key that is used for authentication.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a router or switch:</p> <ul style="list-style-type: none">• The password must be at least eight characters long.• The password can include alphabetic, numeric, and special characters, but it cannot include control characters.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring MD5 Authentication on page 55• Configuring SHA Authentication on page 55

authentication-sha

Syntax	authentication-sha { authentication-password <i>authentication-password</i> ; }
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the secure hash algorithm (SHA) as the authentication type for the SNMPv3 user.




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


The remaining statement is explained separately.

Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring SHA Authentication on page 55

community-name

Syntax	<code>community-name <i>community-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 snmp-community <i>community-index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	The community name defines an SNMP community. The SNMP community authorizes SNMPv1 or SNMPv2 clients. The access privileges associated with the configured security name define which MIB objects are available and the operations (notify, read, or write) allowed on those objects.
Options	<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 (" ").
<div><p>NOTE: Community names must be unique. You cannot configure the same community name at the <code>[edit snmp community]</code> and <code>[edit snmp v3 snmp-community <i>community-index</i>]</code> hierarchy levels.</p><p>The community name at the <code>[edit snmp v3 snmp-community <i>community-index</i>]</code> hierarchy level is encrypted and not displayed in the command-line interface (CLI).</p></div>	
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the SNMPv3 Community on page 79

engine-id

Syntax	engine-id { (local <i>engine-id-suffix</i> use-default-ip-address use-mac-address); }
Hierarchy Level	[edit snmp]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.1 for EX Series switches.
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, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i> . You can configure the suffix here.
	<div>  <p>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 are based on the previous engine ID.</p> <p>For the engine ID, we recommend using the MAC address of fxp0.</p> </div>
Options	<p>local <i>engine-id-suffix</i>—Explicit setting for the engine ID suffix.</p> <p>use-default-ip-address—The engine ID suffix is generated from the default IP address.</p> <p>use-mac-address—The SNMP engine identifier is generated from the MAC address of the management interface on the router.</p> <p>Default: use-default-ip-address</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring the Local Engine ID on page 53

group

See the following sections:

- [group \(Configuring Group Name\)](#) on page 158
- [group \(Defining Access Privileges for an SNMPv3 Group\)](#) on page 159

group (Configuring Group Name)

Syntax

```
group group-name {  
  (default-context-prefix | context-prefix context-prefix){  
    security-model (any | usm | v1 | v2c) {  
      security-level (authentication | none | privacy) {  
        notify-view view-name;  
        read-view view-name;  
        write-view view-name;  
      }  
    }  
  }  
}
```

Hierarchy Level [edit snmp v3 vacm access]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Assign the security name to a group, and specify the SNMPv3 context applicable to the group. The **default-context-prefix** statement, when included, adds all the contexts configured on the device to the group, whereas the **context-prefix *context-prefix*** statement enables you to specify a context and to add that particular context to the group. When the context prefix is specified as default (for example, **context-prefix default**), the context associated with the master routing instance is added to the group.

The remaining statements under this hierarchy are documented in separate topics.

Options *group-name*—SNMPv3 group name created for the SNMPv3 group.

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

Related Documentation

- [Configuring the Group](#) on page 59

group (Defining Access Privileges for an SNMPv3 Group)

Syntax	<code>group group-name;</code>
Hierarchy Level	[edit snmp v3 vacm security-to-group security-model (usm v1 v2c) security-name security-name]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define access privileges granted to a group.
Options	<i>group-name</i> —Identifies a collection of SNMP security names that belong to the same access policy SNMP.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Group on page 64

retry-count

Syntax	<code>retry-count number;</code>
Hierarchy Level	[edit snmp v3 target-address target-address-name]
Release Information	Statement introduced in Junos OS Release 7.4.
Description	Configure the retry count for SNMP informs.
Options	<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. Default: 3 times
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring SNMP Informs on page 75 timeout on page 160

timeout

Syntax	<code>timeout <i>seconds</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 7.4.
Description	Configure the timeout period (in seconds) for SNMP informs.
Options	<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. Default: 15
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring SNMP Informs on page 75• retry-count on page 159

local-engine

Syntax

```
local-engine {
  user username {
    authentication-md5 {
      authentication-password authentication-password;
    }
    authentication-none;
    authentication-sha {
      authentication-password authentication-password;
    }
    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 OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure local engine information for the user-based security model (USM).

The remaining statements are explained separately.

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

Related Documentation

- Creating SNMPv3 Users on page 54

message-processing-model

Syntax	<code>message-processing-model (v1 v2c v3);</code>
Hierarchy Level	<code>[edit snmp v3 target-parameters <i>target-parameter-name</i> parameters]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the message processing model to be used when generating SNMP notifications.
Options	<code>v1</code> —SNMPv1 message process model. <code>v2c</code> —SNMPv2c message process model. <code>v3</code> —SNMPv3 message process model.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Message Processing Model on page 73

notify

Syntax	<pre> notify <i>name</i> { tag <i>tag-name</i>; type (trap inform); } </pre>
Hierarchy Level	[edit snmp v3]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>type inform option added in Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.1 for the QFX Series.</p>
Description	Select management targets for SNMPv3 notifications as well as the type of notifications. Notifications can be either traps or informs.
Options	<p><i>name</i>—Name assigned to the notification.</p> <p><i>tag-name</i>—Notifications are sent to all targets configured with this tag.</p> <p><i>type</i>—Notification type is trap or inform. Traps are unconfirmed notifications. Informs are confirmed notifications.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring the Inform Notification Type and Target Address on page 77 Configuring the SNMPv3 Trap Notification on page 66

notify-filter

See the following sections:

- **notify-filter (Applying to the Management Target)** on page 164
- **notify-filter (Configuring the Profile Name)** on page 164

notify-filter (Applying to the Management Target)

Syntax	<code>notify-filter <i>profile-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-parameters <i>target-parameters-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the notify filter to be used by a specific set of target parameters.
Options	<i>profile-name</i> —Name of the notify filter to apply to notifications.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Applying the Trap Notification Filter on page 73

notify-filter (Configuring the Profile Name)

Syntax	<code>notify-filter <i>profile-name</i> { oid <i>oid</i> (include exclude); }</code>
Hierarchy Level	<code>[edit snmp v3]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Specify a group of MIB objects for which you define access. The notify filter limits the type of traps or informs sent to the network management system.
Options	<i>profile-name</i> —Name assigned to the notify filter. The remaining statement is explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Trap Notification Filter on page 67• oid on page 165

notify-view

Syntax	<code>notify-view <i>view-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 vacm access group <i>group-name</i> (default-context-prefix context-prefix <i>context-prefix</i>) security-model (any usm v1 v2c) security-level (authentication none privacy)]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Associate the notify view with a community (for SNMPv1 or SNMPv2c clients) or a group name (for SNMPv3 clients).
Options	<i>view-name</i> —Name of the view to which the SNMP user group has access.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring MIB Views on page 40 Configuring the Notify View on page 61

oid

Syntax	<code>oid <i>oid</i> (include exclude);</code>
Hierarchy Level	<code>[edit snmp v3 notify-filter <i>profile-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify an object identifier (OID) used to represent a subtree of MIB objects. This OID is a prefix that the represented MIB objects have in common.
Options	exclude —Exclude the subtree of MIB objects represented by the specified OID. include —Include the subtree of MIB objects represented by the specified OID. <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. You can specify the OID using either a sequence of dotted integers or a subtree name.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Trap Notification Filter on page 67

parameters

Syntax	<pre>parameters { message-processing-model (v1 v2c v3); security-level (none authentication privacy); security-model (usm v1 v2c); security-name <i>security-name</i>; }</pre>
Hierarchy Level	[edit snmp v3 target-parameters <i>target-parameters-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure a set of target parameters for message processing and security. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Defining and Configuring the Trap Target Parameters on page 72

port

Syntax	<pre>port <i>port-number</i>;</pre>
Hierarchy Level	[edit snmp v3 target-address <i>target-address-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure a UDP port number for an SNMP target.
Default	If you omit this statement, the default port is 162.
Options	<i>port-number</i> —Port number for the SNMP target.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Port on page 70

privacy-3des

Syntax	<pre>privacy-3des { privacy-password <i>privacy-password</i>; }</pre>
Hierarchy Level	<pre>[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]</pre>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.1 for the QFX Series.</p>
Description	<p>Configure the triple Data Encryption Standard (3DES) as the privacy type for the SNMPv3 user.</p>
Options	<p>privacy-password <i>privacy-password</i>—Password that a user enters. The password is then converted into a key that is used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a router or switch:</p> <ul style="list-style-type: none"> • The password must be at least eight characters long. • The password can include alphabetic, numeric, and special characters, but it cannot include control characters.
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring the Encryption Type on page 56

privacy-aes128

Syntax	<pre>privacy-aes128 { privacy-password <i>privacy-password</i>; }</pre>
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the Advanced Encryption Standard encryption algorithm (CFB128-AES-128 Privacy Protocol) for the SNMPv3 user.
Options	<p>privacy-password <i>privacy-password</i>—Password that a user enters. The password is then converted into a key that is used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a router or switch:</p> <ul style="list-style-type: none">• The password must be at least eight characters long.• The password can include alphabetic, numeric, and special characters, but it cannot include control characters.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Encryption Type on page 56

privacy-des

Syntax	<code>privacy-des { privacy-password <i>privacy-password</i>; }</code>
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the Data Encryption Standard (DES) as the privacy type for the SNMPv3 user.
Options	<p>privacy-password <i>privacy-password</i>—Password that a user enters. The password is then converted into a key that is used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a router or switch:</p> <ul style="list-style-type: none"> • The password must be at least eight characters long. • The password can include alphabetic, numeric, and special characters, but it cannot include control characters.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Configuring the Encryption Type on page 56

privacy-none

Syntax	<code>privacy-none;</code>
Hierarchy Level	[edit snmp v3 usm local-engine user <i>username</i>], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure that no encryption be used for the SNMPv3 user.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Configuring the Encryption Type on page 56

privacy-password

Syntax	<code>privacy-password <i>privacy-password</i>;</code>
Hierarchy Level	[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], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> privacy-3des], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> privacy-aes128], [edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> privacy-des]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure a privacy password for the SNMPv3 user.
Options	<p><i>privacy-password</i>—Password that a user enters. The password is then converted into a key that is used for encryption.</p> <p>SNMPv3 has special requirements when you create plain-text passwords on a router or switch:</p> <ul style="list-style-type: none">• The password must be at least eight characters long.• The password can include alphabetic, numeric, and special characters, but it cannot include control characters.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Encryption Type on page 56

read-view

Syntax	<code>read-view <i>view-name</i>;</code>
Hierarchy Level	[<code>edit snmp v3 vacm access group <i>group-name</i> (default-context-prefix context-prefix <i>context-prefix</i>) security-model (any usm v1 v2c) security-level (authentication none privacy)</code>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Associate the read-only view with a community (for SNMPv1 or SNMPv2c clients) or a group name (for SNMPv3 clients).
Options	<i>view-name</i> —The name of the view to which the SNMP user group has access.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Read View on page 61Configuring MIB Views on page 40

remote-engine

Syntax	<pre>remote-engine <i>engine-id</i> { user <i>username</i> { authentication-md5 { authentication-password <i>authentication-password</i>; } authentication-none; authentication-sha { authentication-password <i>authentication-password</i>; } privacy-aes128 { privacy-password <i>privacy-password</i>; } privacy-des { privacy-password <i>privacy-password</i>; } privacy-3des { privacy-password <i>privacy-password</i>; } privacy-none { privacy-password <i>privacy-password</i>; } } }</pre>
Hierarchy Level	[edit snmp v3 usm]
Release Information	Statement introduced in Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the 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.
Options	<i>engine-id</i> —Engine identifier. Used to compute the security digest for authenticating and encrypting packets sent to a user on the remote host. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Remote Engine and Remote User on page 76

routing-instance

Syntax	<code>routing-instance <i>routing-instance-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-address <i>target-address-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.3. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify a routing instance for an SNMPv3 trap target.
Options	<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, test-ls/test-ri). To configure the default routing instance on a logical system, specify the logical system name followed by default (for example, test-ls/default).</p>
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Trap Target Address on page 68

security-level

See the following sections:

- **security-level (Defining Access Privileges)** on page 174
- **security-level (Generating SNMP Notifications)** on page 175

security-level (Defining Access Privileges)

Syntax	<pre>security-level (authentication none privacy) { notify-view <i>view-name</i>; read-view <i>view-name</i>; write-view <i>view-name</i>; }</pre>
Hierarchy Level	[edit snmp v3 vacm access group <i>group-name</i> (default-context-prefix context-prefix <i>context-prefix</i>) security-model (any usm v1 v2c)]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Define the security level used for access privileges.
Default	none
Options	authentication —Provide authentication but no encryption. none —No authentication and no encryption. privacy —Provide authentication and encryption.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Security Level on page 60

security-level (Generating SNMP Notifications)

Syntax	<code>security-level (authentication none privacy);</code>
Hierarchy Level	<code>[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the security level to use when generating SNMP notifications.
Default	<code>none</code>
Options	authentication —Provide authentication but no encryption. none —No authentication and no encryption. privacy —Provide authentication and encryption.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Security Level on page 74

security-model

See the following sections:

- **security-model (Access Privileges)** on page 176
- **security-model (Group)** on page 177
- **security-model (SNMP Notifications)** on page 177

security-model (Access Privileges)

Syntax	security-model (usm v1 v2c);
Hierarchy Level	[edit snmp v3 vacm access group <i>group-name</i> (default-context-prefix context-prefix <i>context-prefix</i>)]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the security model for an SNMPv3 group. The security model is used to determine access privileges for the group.
Options	usm —SNMPv3 security model. v1 —SNMPv1 security model. v2c —SNMPv2c security model.
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Security Model on page 60

security-model (Group)

Syntax	security-model (usm v1 v2c) { security-name <i>security-name</i> { group <i>group-name</i> ; } }
Hierarchy Level	[edit snmp v3 vacm security-to-group]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Define a security model for a group.
Options	usm—SNMPv3 security model. v1—SNMPv1 security model. v2c—SNMPv2c security model.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Security Model on page 63

security-model (SNMP Notifications)

Syntax	security-model (usm v1 v2c);
Hierarchy Level	[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the security model for an SNMPv3 group. The security model is used for SNMP notifications.
Options	usm—SNMPv3 security model. v1—SNMPv1 security model. v2c—SNMPv2c security model.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Security Model on page 73

security-name

See the following sections:

- **security-name (Community String)** on page 178
- **security-name (Security Group)** on page 179
- **security-name (SNMP Notifications)** on page 180

security-name (Community String)

Syntax	<code>security-name <i>security-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 snmp-community <i>community-index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Associate the community string configured at the <code>[edit snmp v3 snmp-community <i>community-index</i>]</code> hierarchy level to a security name.
Options	<i>security-name</i> —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.

Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Security Names on page 81

security-name (Security Group)

Syntax	<code>security-name <i>security-name</i> { group <i>group-name</i>; }</code>
Hierarchy Level	[edit snmp v3 vacm security-to-group security-model (usm v1 v2c)]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Associate a group or a community string with a configured security group.
Options	<i>security-name</i> —Username configured at the [edit snmp v3 usm local-engine user <i>username</i>] hierarchy level. For SNMPv1 and SNMPv2c, the security name is the community string configured at the [edit snmp v3 snmp-community <i>community-index</i>] hierarchy level.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Assigning Security Names to Groups on page 64

security-name (SNMP Notifications)

Syntax	<code>security-name <i>security-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the security name used when generating SNMP notifications.
Options	<i>security-name</i> —If the SNMPv3 USM security model is used, identify the user when generating the SNMP notification. If the v1 or v2c security models are used, identify the SNMP community used when generating the notification.
<div> NOTE: 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 <i>community-index</i>]</code> hierarchy level.</p>	
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Security Name on page 74

security-to-group

Syntax	<pre>security-to-group { security-model (usm v1 v2c) { group <i>group-name</i>; security-name <i>security-name</i>; } }</pre>
Hierarchy Level	[edit snmp v3 vacm]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.1 for the QFX Series.</p>
Description	<p>Configure the group to which a specific SNMPv3 security name belongs. The security name is used for messaging security.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Assigning Security Model and Security Name to a Group on page 63

snmp-community

Syntax	<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>
Hierarchy Level	[edit snmp v3]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	Configure the SNMP community.
Options	<p><i>community-index</i>—(Optional) String that identifies an SNMP community.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring the SNMPv3 Community on page 79

tag

Syntax	<code>tag tag-name;</code>
Hierarchy Level	[edit snmp v3 notify <i>name</i>], [edit snmp v3 snmp-community <i>community-index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure a set of targets to receive traps or informs (for IPv4 packets only).
Options	<i>tag-name</i> —Identifies the address of managers that are allowed to use a community string.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Tag on page 81Configuring the SNMPv3 Trap Notification on page 66

tag-list

Syntax	<code>tag-list tag-list;</code>
Hierarchy Level	[edit snmp v3 target-address <i>target-address-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure an SNMP tag list used to select target addresses.
Options	<i>tag-list</i> —Define sets of target addresses (tags). To specify more than one tag, specify the tag names as a space-separated list enclosed within double quotes.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Trap Target Address on page 70

target-address

Syntax	<pre>target-address <i>target-address-name</i> { address <i>address</i>; address-mask <i>address-mask</i>; logical-system <i>logical-system</i>; port <i>port-number</i>; retry-count <i>number</i>; routing-instance <i>instance</i>; tag-list <i>tag-list</i>; target-parameters <i>target-parameters-name</i>; timeout <i>seconds</i>; }</pre>
Hierarchy Level	[edit snmp v3]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	Configure the address of an SNMP management application and the parameters to be used in sending notifications.
Options	<p><i>target-address-name</i>—String that identifies the target address.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring the Trap Target Address on page 68

target-parameters

Syntax At the **[edit snmp v3]** hierarchy level:

```
target-parameters target-parameters-name {  
  profile-name;  
  parameters {  
    message-processing-model (v1 | v2c | V3);  
    security-level (authentication | none | privacy);  
    security-model (usm | v1 | v2c);  
    security-name security-name;  
  }  
}
```

At the **[edit snmp v3 target-address *target-address-name*]** hierarchy level:

```
target-parameters target-parameters-name;
```

Hierarchy Level [edit snmp v3]
[edit snmp v3 target-address *target-address-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the message processing and security parameters for sending notifications to a particular management target. The target parameters are configured at the **[edit snmp v3]** hierarchy level. The remaining statements at this level are explained separately.

Then apply the target parameters configured at the **[edit snmp v3 target-parameters *target-parameters-name*]** hierarchy level to the target address configuration at the **[edit snmp v3]** hierarchy level.

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

Related Documentation

- Defining and Configuring the Trap Target Parameters on page 72
- Applying Target Parameters on page 71

type

Syntax	<code>type (inform trap);</code>
Hierarchy Level	<code>[edit snmp v3 notify <i>name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. inform option added in Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the type of SNMP notification.
Options	inform —Defines the type of notification as an inform. SNMP informs are confirmed notifications. trap —Defines the type of notification as a trap. SNMP traps are unconfirmed notifications.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring SNMP Informs on page 75 Configuring the SNMPv3 Trap Notification on page 66

user

Syntax	<code>user <i>username</i>;</code>
Hierarchy Level	<code>[edit snmp v3 usm local-engine],</code> <code>[edit snmp v3 usm remote-engine <i>engine-id</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Specify a user associated with an SNMPv3 group on a local or remote SNMP engine.
Options	<i>username</i> —SNMPv3 user-based security model (USM) username.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Creating SNMPv3 Users on page 54

usm

```

Syntax  usm {
        local-engine {
            user username {
                authentication-md5 {
                    authentication-password authentication-password;
                }
                authentication-none;
                authentication-sha {
                    authentication-password authentication-password;
                }
                privacy-aes128 {
                    privacy-password privacy-password;
                }
                privacy-des {
                    privacy-password privacy-password;
                }
                privacy-3des {
                    privacy-password privacy-password;
                }
                privacy-none {
                    privacy-password privacy-password;
                }
            }
        }
        remote-engine engine-id {
            user username {
                authentication-md5 {
                    authentication-password authentication-password;
                }
                authentication-none;
                authentication-sha {
                    authentication-password authentication-password;
                }
                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]

Release Information Statement introduced before Junos OS Release 7.4.

Statement introduced in Junos OS Release 9.0 for EX Series switches.

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure user-based security model (USM) information.

The remaining statements are explained separately.

Required Privilege snmp—To view this statement in the configuration.

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

- Related Documentation**
- Creating SNMPv3 Users on page 54
 - Configuring the Remote Engine and Remote User on page 76

v3

```

Syntax  v3 {
    notify name {
        tag tag-name;
        type trap;
    }
    notify-filter profile-name {
        oid object-identifier (include | exclude);
    }
    snmp-community community-index {
        community-name community-name;
        security-name security-name;
        tag tag-name;
    }
    target-address target-address-name {
        address address;
        address-mask address-mask;
        logical-system logical-system;
        port port-number;
        retry-count number;
        routing-instance instance;
        tag-list tag-list;
        target-parameters target-parameters-name;
        timeout seconds;
    }
    target-parameters target-parameters-name {
        notify-filter profile-name;
        parameters {
            message-processing-model (v1 | v2c | V3);
            security-level (authentication | none | privacy);
            security-model (usm | v1 | v2c);
            security-name security-name;
        }
    }
    usm {
        local-engine {
            user username {
                authentication-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 | context-prefix context-prefix) {
                security-model (any | usm | v1 | v2c) {
                    security-level (authentication | none | privacy) {
                        notify-view view-name;
                        read-view view-name;
                        write-view view-name;
                    }
                }
            }
        }
    }
}
security-to-group {
    security-model (usm | v1 | v2c) {
        security-name security-name {
            group group-name;
        }
    }
}
}
}

```

Hierarchy Level [edit snmp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description	Configure SNMPv3. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Minimum SNMPv3 Configuration on a Device Running Junos OS on page 52

vacm

Syntax	<pre>vacm { access { group <i>group-name</i> { (default-context-prefix context-prefix <i>context-prefix</i>){ security-model (any usm v1 v2c) { security-level (authentication none privacy) { notify-view <i>view-name</i>; read-view <i>view-name</i>; write-view <i>view-name</i>; } } } } } security-to-group { security-model (usm v1 v2c); security-name <i>security-name</i> { group <i>group-name</i>; } } }</pre>
Hierarchy Level	[edit snmp v3]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure view-based access control model (VACM) information. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Defining Access Privileges for an SNMP Group on page 58

view

See [view \(Configuring a MIB View\)](#).

write-view

Syntax	<code>write-view view-name;</code>
Hierarchy Level	[edit snmp v3 vacm access group <i>group-name</i> (default-context-prefix context-prefix <i>context-prefix</i>) security-model (any usm v1 v2c) security-level (authentication none privacy)]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series switches.
Description	Associate the write view with a community (for SNMPv1 or SNMPv2c clients) or a group name (for SNMPv3 clients).
Options	<i>view-name</i> —Name of the view for which the SNMP user group has write permission.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring MIB Views on page 40• Configuring the Write View on page 62

PART 4

RMON Alarms and Events

- [Configuring RMON Alarms and Events on page 195](#)
- [Monitoring RMON Alarms and Events on page 203](#)
- [Summary of RMON Alarm and Event Configuration Statements on page 213](#)

Configuring RMON Alarms and Events

This chapter contains the following topics:

- Understanding RMON Alarms and Events Configuration on page 195
- Minimum RMON Alarm and Event Entry Configuration on page 196
- Configuring an Alarm Entry and Its Attributes on page 196
- Configuring an Event Entry and Its Attributes on page 200
- Example: Configuring an RMON Alarm and Event Entry on page 201

Understanding RMON Alarms and Events Configuration

Junos OS 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 MIB object.

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;
    }
  }
}
```

- Related Documentation**
- Understanding RMON Alarms on page 203
 - Understanding RMON Events on page 208
 - Configuring an Alarm Entry and Its Attributes on page 196
 - Configuring an Event Entry and Its Attributes on page 200
 - Using alarmTable to Monitor MIB Objects on page 205
 - Using eventTable to Log Alarms on page 209
 - Minimum RMON Alarm and Event Entry Configuration on page 196

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;
```

- Related Documentation**
- Understanding RMON Alarms and Events Configuration on page 195
 - Configuring an Alarm Entry and Its Attributes on page 196
 - Configuring an Event Entry and Its Attributes on page 200

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 197
- Configuring the Description on page 197
- Configuring the Falling Event Index or Rising Event Index on page 197
- Configuring the Falling Threshold or Rising Threshold on page 198
- Configuring the Interval on page 198
- Configuring the Falling Threshold Interval on page 198
- Configuring the Request Type on page 199
- Configuring the Sample Type on page 199
- Configuring the Startup Alarm on page 199

- Configuring the System Log Tag on page 200
- Configuring the Variable on page 200

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**.

Related Documentation

- Understanding RMON Alarms and Events Configuration on page 195
- Understanding RMON Alarms on page 203
- Understanding RMON Events on page 208
- Configuring an Alarm Entry and Its Attributes on page 196
- Minimum RMON Alarm and Event Entry Configuration on page 196
- Example: Configuring an RMON Alarm and Event Entry on page 201

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;
  }
}
```

Related Documentation

- Understanding RMON Alarms and Events Configuration on page 195
- Configuring an Alarm Entry and Its Attributes on page 196

- [Configuring an Event Entry and Its Attributes on page 200](#)

Monitoring RMON Alarms and Events

Use the remote monitoring (RMON) alarms and events feature to monitor integer-valued MIB objects, standard or enterprise-specific, on a Juniper Networks router. Configuration and operational information are in the MIB objects defined in **alarmTable**, **eventTable**, and **logTable** in RFC 2819. Additional information is defined by the Juniper Networks enterprise-specific extension to **alarmTable** defined in **jnxRmonMIB** (**jnx-rmon-mib.txt**).

This chapter covers the following main topics:

- Understanding RMON Alarms on page 203
- Using alarmTable to Monitor MIB Objects on page 205
- Understanding RMON Events on page 208
- Using eventTable to Log Alarms on page 209

Understanding RMON Alarms

An RMON alarm identifies:

- A specific MIB object that is monitored.
- The frequency of sampling.
- 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 204
- jnxRmonAlarmTable on page 204

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, the associated event alarm does not take any action.

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 http://www.juniper.net/techpubs/en_US/junos10.3/topics/reference/mibs/mib-jnx-rmon.txt.

For more information about the Juniper Networks enterprise-specific extensions to the RMON Events and Alarms MIB, see “RMON Events and Alarms MIB” in the *Junos OS SNMP MIBs and Traps Reference*.

Related Documentation

- Understanding RMON Events on page 208
- Configuring an Alarm Entry and Its Attributes on page 196
- Using alarmTable to Monitor MIB Objects on page 205

Using alarmTable to Monitor MIB Objects

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

- Creating an Alarm Entry on page 205
- Configuring the Alarm MIB Objects on page 205
- Activating a New Row in alarmTable on page 207
- Modifying an Active Row in alarmTable on page 207
- Deactivating a Row in alarmTable on page 208

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 206
- alarmVariable on page 206
- alarmSampleType on page 206
- alarmValue on page 206
- alarmStartupAlarm on page 206
- alarmRisingThreshold on page 206
- alarmFallingThreshold on page 207
- alarmOwner on page 207
- alarmRisingEventIndex on page 207
- alarmFallingEventIndex on page 207

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 command-line interface (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
```

Related Documentation

- Understanding RMON Alarms on page 203
- Understanding RMON Events on page 208
- Configuring an Alarm Entry and Its Attributes on page 196

Understanding 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 208

eventTable

eventTable contains the following objects:

- **eventIndex**—An index that uniquely identifies an entry in **eventTable**. Each entry defines one event that is 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 command-line interface (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) are deleted.

Related Documentation

- Understanding RMON Alarms on page 203
- Configuring an Event Entry and Its Attributes on page 200
- Using eventTable to Log Alarms on page 209

Using eventTable to Log Alarms

To use **eventTable** to log alarms, perform the following tasks:

- Creating an Event Entry on page 209
- Configuring the MIB Objects on page 209
- Activating a New Row in eventTable on page 211
- Deactivating a Row in eventTable on page 211

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:



NOTE: The **eventType** object is required. All other objects are optional.

- eventType on page 209
- eventCommunity on page 210
- eventOwner on page 210
- eventDescription on page 210

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 command-line interface (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 command-line interface (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
```

Related Documentation

- Understanding RMON Alarms on page 203
- Understanding RMON Events on page 208
- Configuring an Event Entry and Its Attributes on page 200

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

Hierarchy Level [edit snmp rmon]

Release Information Statement introduced before Junos OS Release 7.4.
 Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Configure RMON alarm entries.

Options *index*—Identifies this alarm entry as an integer.

The remaining statements are explained separately.

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

Related Documentation • Configuring an Alarm Entry and Its Attributes on page 196
 • event on page 215

community

Syntax	<code>community <i>community-name</i>;</code>
Hierarchy Level	[edit snmp rmon event <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set.
Options	community-name —Identifies the trap group that is used when generating a trap if the event is configured to send traps.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring an Event Entry and Its Attributes on page 200

description

Syntax	<code>description <i>description</i>;</code>
Hierarchy Level	[edit snmp rmon alarm <i>index</i>], [edit snmp rmon event <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Text description of alarm or event.
Options	description —Text description of an alarm or event entry. If the description includes spaces, enclose it in quotation marks (" ").
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Description on page 197Configuring an Event Entry and Its Attributes on page 200

event

Syntax	<pre>event <i>index</i> { community <i>community-name</i>; description <i>description</i>; type <i>type</i>; }</pre>
Hierarchy Level	[edit snmp rmon]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure RMON event entries.
Options	<i>index</i> —Identifier for a specific event entry. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring an Event Entry and Its Attributes on page 200 alarm on page 213

falling-event-index

Syntax	falling-event-index <i>index</i> ;
Hierarchy Level	[edit snmp rmon alarm <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	The index of the event entry that is used when a falling threshold is crossed. If this value is zero, no event is triggered.
Options	<i>index</i> —Index of the event entry that is used when a falling threshold is crossed. Range: 0 through 65,535 Default: 0
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Falling Event Index or Rising Event Index on page 197 rising-event-index on page 219

falling-threshold

Syntax	<code>falling-threshold <i>integer</i>;</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	The lower threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup-alarm value is equal to falling-alarm value or rising-or-falling-alarm value. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising-threshold .
Options	<i>integer</i> —The lower threshold for the alarm entry. Range: -2,147,483,648 through 2,147,483,647 Default: 20 percent less than rising-threshold
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Falling Threshold or Rising Threshold on page 198rising-threshold on page 219

falling-threshold-interval

Syntax	<code>falling-threshold-interval <i>seconds</i>;</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.3. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Interval between samples when the rising threshold is crossed. Once the alarm crosses the falling threshold, the regular sampling interval is used.
Options	<i>seconds</i> —Time between samples, in seconds. Range: 1 through 2,147,483,647 seconds Default: 60 seconds
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Falling Threshold Interval on page 198interval on page 217

interval

Syntax	<code>interval <i>seconds</i>;</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Interval between samples.
Options	<i>seconds</i> —Time between samples, in seconds. Range: 1 through 2,147,483,647 seconds Default: 60 seconds
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Interval on page 198

request-type

Syntax	<code>request-type (get-next-request get-request walk-request);</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.3. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Extend monitoring to a specific SNMP object instance (get-request), or extend monitoring to all object instances belonging to a MIB branch (walk-request), or extend monitoring to the next object instance after the instance specified in the configuration (get-next-request).
Options	get-next-request —Performs an SNMP get next request. get-request —Performs an SNMP get request. walk-request —Performs an SNMP walk request. Default: walk-request
Required Privilege Level	snmp —To view this statement in the configuration. snmp-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Request Type on page 199variable on page 222

rising-event-index

Syntax	<code>rising-event-index <i>index</i>;</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Index of the event entry that is used when a rising threshold is crossed. If this value is zero, no event is triggered.
Options	<i>index</i> —Index of the event entry that is used when a rising threshold is crossed. Range: 0 through 65,535 Default: 0
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Falling Event Index or Rising Event Index on page 197falling-event-index on page 215

rising-threshold

Syntax	<code>rising-threshold <i>integer</i>;</code>
Hierarchy Level	<code>[edit snmp rmon alarm <i>index</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	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 value is equal to the falling alarm or rising or falling alarm value. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling threshold.
Options	<i>integer</i> —The lower threshold for the alarm entry. Range: -2,147,483,648 through 2,147,483,647
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Falling Threshold or Rising Threshold on page 198falling-threshold on page 216

rmon

Syntax	<code>rmon { ... }</code>
Hierarchy Level	<code>[edit snmp]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure Remote Monitoring.
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring an Alarm Entry and Its Attributes on page 196

sample-type

Syntax	<code>sample-type (absolute-value delta-value);</code>
Hierarchy Level	<code>[edit snmp rmon alarm index]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Method of sampling the selected variable.
Options	<p>absolute-value—Actual value of the selected variable is used when comparing against the thresholds.</p> <p>delta-value—Difference between samples of the selected variable is used when comparing against the thresholds.</p>
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Sample Type on page 199

startup-alarm

Syntax	startup-alarm (falling-alarm rising-alarm rising-or-falling-alarm);
Hierarchy Level	[edit snmp rmon alarm <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	The alarm that can be sent upon entry startup.
Options	<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>Default: rising-or-falling-alarm</p>
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Startup Alarm on page 199

syslog-subtag

Syntax	syslog-subtag <i>syslog-subtag</i> ;
Hierarchy Level	[edit snmp rmon alarm <i>index</i>]
Release Information	Statement introduced in Junos OS Release 8.5. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Add a tag to the system log message.
Options	<p>syslog-subtag <i>syslog-subtag</i>—Tag of not more than 80 uppercase characters to be added to syslog messages.</p> <p>Default: None</p>
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the System Log Tag on page 200

type

Syntax	<code>type type;</code>
Hierarchy Level	[edit snmp rmon event <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Type of notification generated when a threshold is crossed.
Options	type —Type of notification: <ul style="list-style-type: none">• log—Add an entry to logTable.• log-and-trap—Send an SNMP trap and make a log entry.• none—No notifications are sent.• snmptrap—Send an SNMP trap. Default: log-and-trap
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring an Event Entry and Its Attributes on page 200

variable

Syntax	<code>variable oid-variable;</code>
Hierarchy Level	[edit snmp rmon alarm <i>index</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Object identifier (OID) of MIB variable to be monitored.
Options	oid-variable —OID of the MIB variable that is being monitored. The OID can be a dotted decimal (for example, 1.3.6.1.2.1.2.1.2.1.10.1). Alternatively, use the MIB object name (for example, ifInOctets.1).
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring the Variable on page 200

PART 5

Health Monitoring

- [Configuring Health Monitoring on page 225](#)
- [Summary of Health Monitoring Configuration Statements on page 229](#)

CHAPTER 16

Configuring Health Monitoring

This chapter contains the following topics:

- Configuring Health Monitoring on Devices Running Junos OS on page 225
- Example: Configuring Health Monitoring on page 228

Configuring Health Monitoring on Devices Running Junos OS

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 routers, RMON alarms and events provide much of the infrastructure needed to reduce the polling overhead from the NMS. 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 OS processes).

Health monitoring is designed to minimize user configuration requirements. To configure health monitoring entries, include the **health-monitor** statement at the **[edit snmp]** hierarchy level:

```
[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 226
- Minimum Health Monitoring Configuration on page 227
- Configuring the Falling Threshold or Rising Threshold on page 227
- Configuring the Interval on page 227
- Log Entries and Traps on page 228

Monitored Objects

When you configure the health monitor, monitoring information for certain object instances is available, as shown in Table 14 on page 226.

Table 14: Monitored Object Instances

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

Minimum Health Monitoring Configuration

To enable health monitoring on the router or switch, 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.

Related Documentation

- Understanding RMON Alarms and Events Configuration on page 195
- Configuring an Alarm Entry and Its Attributes on page 196
- Configuring an Event Entry and Its Attributes on page 200
- Example: Configuring Health Monitoring on page 228
- Understanding Device Management Functions in Junos OS on page 3

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.

Related Documentation

- Configuring Health Monitoring on Devices Running Junos OS on page 225

CHAPTER 17

Summary of Health Monitoring Configuration Statements

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

falling-threshold

Syntax	<code>falling-threshold <i>percentage</i>;</code>
Hierarchy Level	<code>[edit snmp]</code>
Release Information	Statement introduced in Junos OS Release 8.0. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	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 rising-threshold .
Options	<i>percentage</i> —The lower threshold for the alarm entry. Range: 1 through 100 Default: 70 percent of the maximum possible value
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Falling Threshold or Rising Threshold on page 227rising-threshold on page 231

health-monitor

Syntax	health-monitor { falling-threshold <i>percentage</i> ; interval <i>seconds</i> ; rising-threshold <i>percentage</i> ; }
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 8.0. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure health monitoring. The remaining statements are explained separately.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring Health Monitoring on Devices Running Junos OS on page 225

interval

Syntax	interval <i>seconds</i> ;
Hierarchy Level	[edit snmp health-monitor]
Release Information	Statement introduced in Junos OS Release 8.0. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Interval between samples.
Options	seconds —Time between samples, in seconds. Range: 1 through 2147483647 seconds Default: 300 seconds
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Interval on page 227

rising-threshold

Syntax	<code>rising-threshold <i>percentage</i>;</code>
Hierarchy Level	<code>[edit snmp]</code>
Release Information	Statement introduced in Junos OS Release 8.0. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	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 falling-threshold .
Options	<i>percentage</i> —The lower threshold for the alarm entry. Range: 1 through 100 Default: 80 percent of the maximum possible value
Required Privilege Level	<code>snmp</code> —To view this statement in the configuration. <code>snmp-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• falling-threshold on page 229• Configuring the Falling Threshold or Rising Threshold on page 227

PART 6

Monitoring Service Quality

- [Monitoring Service Quality in Service Provider Networks on page 235](#)

Monitoring Service Quality in Service Provider Networks

This chapter includes the following topics:

- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235
- Understanding RMON for Monitoring Service Quality on page 237
- Defining and Measuring Network Availability on page 241
- Measuring Health on page 246
- Measuring Performance on page 253

Understanding Measurement Points, Key Performance Indicators, and Baseline Values

This chapter topic 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. You should have a thorough understanding of the SNMP and the associated MIB supported by Junos OS.



NOTE: For a good introduction to the process of monitoring an IP network, see RFC 2330, *Framework for IP Performance Metrics*.

This topic contains the following sections:

- Measurement Points on page 235
- Basic Key Performance Indicators on page 236
- Setting Baselines on page 237

Measurement Points

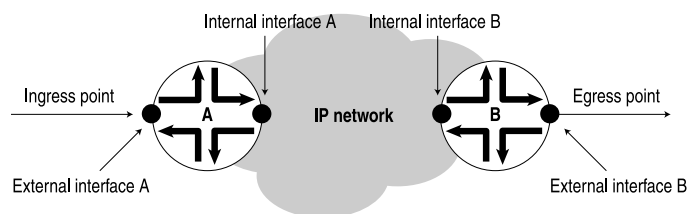
Defining the measurement points where metrics are measured is equally as important as defining the metrics themselves. This section describes measurement points within the context of this chapter and helps identify where measurements can be taken from a service provider network. It is important to understand 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 236.

- 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



NOTE: Figure 3 on page 236 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.

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 must 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.

Related Documentation

- Understanding RMON for Monitoring Service Quality on page 237
- Defining and Measuring Network Availability on page 241
- Measuring Health on page 246
- Measuring Performance on page 253

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*.

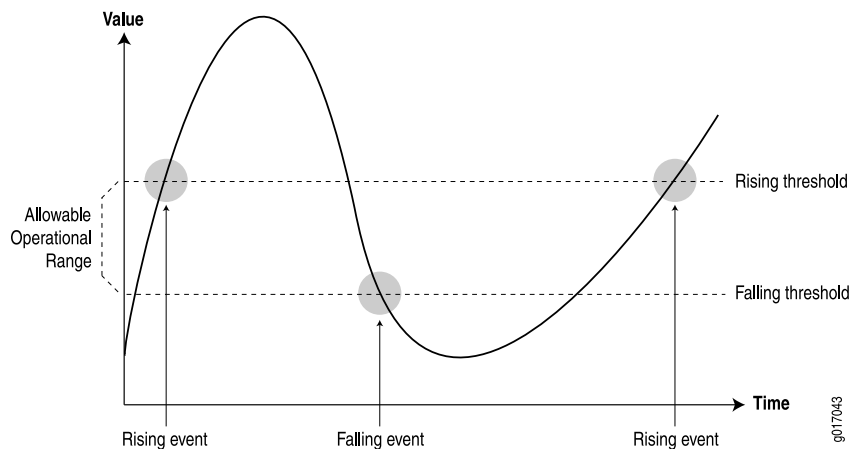
This topic includes the following sections:

- Setting Thresholds on page 237
- RMON Command-Line Interface on page 238
- RMON Event Table on page 239
- RMON Alarm Table on page 239
- Troubleshooting RMON on page 240

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 238.)

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 length of 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

Junos OS 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 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 15 on page 239.) 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 15: 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 OS 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 16 on page 239.

Table 16: RMON Alarm Table

Field	Description
alarmStatus	Status of this row (for example, valid , invalid , or createRequest)

Table 16: RMON Alarm Table (*continued*)

Field	Description
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 (absolute or delta changes)
alarmStartupAlarm	Initial alarm (rising , falling , or either)
alarmRisingThreshold	Rising threshold against which to compare the value
alarmFallingThreshold	Falling threshold against which to compare the value
alarmRisingEventIndex	Index (row) of the rising event in the event table
alarmFallingEventIndex	Index (row) of the falling event in the event table

Both the **alarmStatus** and **eventStatus** fields are **entryStatus** primitives, as defined in RFC 2579, *Textual Conventions for SMIV2*.

Troubleshooting RMON

You troubleshoot the RMON agent, **rmopd**, that runs on the router by inspecting the contents of the Juniper Networks enterprise RMON MIB, **jnxRmon**, which provides the extensions listed in Table 17 on page 240 to the RFC 2819 **alarmTable**.

Table 17: jnxRmon Alarm Extensions

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

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

Related Documentation

- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235

- Defining and Measuring Network Availability on page 241
- Measuring Health on page 246
- Measuring Performance on page 253

Defining and Measuring Network Availability

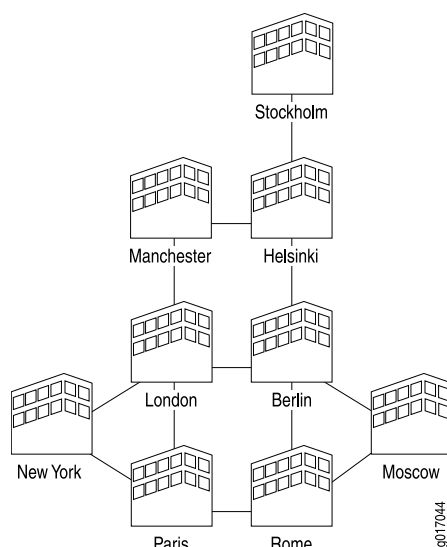
This topic includes the following sections:

- Defining Network Availability on page 241
- Measuring Availability on page 243

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 241.

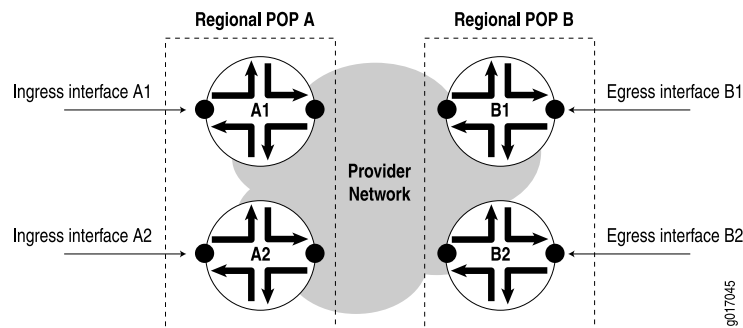
Figure 5: Regional Points of Presence



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 242.

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 242, 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:

$[n \times (n-1)] / 2$ gives $[68 \times (68-1)] / 2 = 2278$ 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:

$[n \times (n-1)] / 2$ gives $[24 \times (24-1)] / 2 = 276$ 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 18 on page 244.

Table 18: Real-Time Performance Monitoring Configuration Options

Field	Description
Request Information	
Probe Type	Type of probe to send as part of the test. Probe types can be: <ul style="list-style-type: none"> • http-get • http-get-metadata • icmp-ping • icmp-ping-timestamp • tcp-ping • udp-ping
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.
Maximum Probe Thresholds	
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.

Table 18: Real-Time Performance Monitoring Configuration Options (*continued*)

Field	Description
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 router, 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** command-line interface (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
```

Related Documentation

- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235
- Understanding RMON for Monitoring Service Quality on page 237
- Measuring Health on page 246
- Measuring Performance on page 253

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 19 on page 246.

Table 19: Health Metrics

Metric:	Errors in
Description	Number of inbound packets that contained errors, preventing them from being delivered
MIB name	IF-MIB (RFC 2233)
Variable name	ifInErrors
Variable OID	.1.3.6.1.31.2.2.1.14
Frequency (mins)	60

Table 19: Health Metrics (*continued*)

Allowable range	To be baselined
Managed objects	Logical interfaces
Metric:	Errors out
Description	Number of outbound packets that contained errors, preventing them from being transmitted
MIB name	IF-MIB (RFC 2233)
Variable name	ifOutErrors
Variable OID	.1.3.6.1.31.2.2.1.20
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	Logical interfaces
Metric:	Discards in
Description	Number of inbound packets discarded, even though no errors were detected
MIB name	IF-MIB (RFC 2233)
Variable name	ifInDiscards
Variable OID	.1.3.6.1.31.2.2.1.13
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	Logical interfaces
Metric:	Unknown protocols
Description	Number of inbound packets discarded because they were of an unknown protocol
MIB name	IF-MIB (RFC 2233)
Variable name	ifInUnknownProtos
Variable OID	.1.3.6.1.31.2.2.1.15

Table 19: Health Metrics (*continued*)

Frequency (mins)	60
Allowable range	To be baselined
Managed objects	Logical interfaces
Metric:	Interface operating status
Description	Operational status of an interface
MIB name	IF-MIB (RFC 2233)
Variable name	ifOperStatus
Variable OID	.1.3.6.1.31.2.2.1.8
Frequency (mins)	15
Allowable range	1 (up)
Managed objects	Logical interfaces
Metric:	Label Switched Path (LSP) state
Description	Operational state of an MPLS label-switched path
MIB name	MPLS-MIB
Variable name	mplsLspState
Variable OID	mplsLspEntry.2
Frequency (mins)	60
Allowable range	2 (up)
Managed objects	All label-switched paths in the network
Metric:	Component operating status
Description	Operational status of a router hardware component
MIB name	JUNIPER-MIB
Variable name	jnxOperatingState
Variable OID	.1.3.6.1.4.1.2636.1.13.1.6
Frequency (mins)	60

Table 19: Health Metrics (*continued*)

Allowable range	2 (running) or 3 (ready)
Managed objects	All components in each Juniper Networks router
Metric:	Component operating temperature
Description	Operational temperature of a hardware component, in Celsius
MIB name	JUNIPER-MIB
Variable name	jnxOperatingTemp
Variable OID	.1.3.6.1.4.1.2636.1.13.1.7
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All components in a chassis
Metric:	System up time
Description	Time, in milliseconds, that the system has been operational.
MIB name	MIB-2 (RFC 1213)
Variable name	sysUpTime
Variable OID	.1.3.6.1.1.3
Frequency (mins)	60
Allowable range	Increasing only (decrement indicates a restart)
Managed objects	All routers
Metric:	No IP route errors
Description	Number of packets that could not be delivered because there was no IP route to their destination.
MIB name	MIB-2 (RFC 1213)
Variable name	ipOutNoRoutes
Variable OID	ip.12
Frequency (mins)	60

Table 19: Health Metrics (*continued*)

Allowable range	To be baselined
Managed objects	Each router
Metric:	Wrong SNMP community names
Description	Number of incorrect SNMP community names received
MIB name	MIB-2 (RFC 1213)
Variable name	snmpInBadCommunityNames
Variable OID	snmp.4
Frequency (hours)	24
Allowable range	To be baselined
Managed objects	Each router
Metric:	SNMP community violations
Description	Number of valid SNMP communities used to attempt invalid operations (for example, attempting to perform SNMP Set requests)
MIB name	MIB-2 (RFC 1213)
Variable name	snmpInBadCommunityUses
Variable OID	snmp.5
Frequency (hours)	24
Allowable range	To be baselined
Managed objects	Each router
Metric:	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

Table 19: Health Metrics (*continued*)

Allowable range	To be baselined
Managed objects	All Juniper Networks routers with redundant Routing Engines
Metric:	FRU state
Description	Operational status of each field-replaceable unit (FRU)
MIB name	JUNIPER-MIB
Variable name	jnxFruState
Variable OID	jnxFruEntry.8
Frequency (mins)	15
Allowable range	2 through 6 for ready/online states. See jnxFruOfflineReason in the event of a FRU failure.
Managed objects	All FRUs in all Juniper Networks routers.
Metric:	Rate of 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.
Metric:	Interface utilization: octets received
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

Table 19: Health Metrics (*continued*)

Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All operational interfaces in the network
Metric:	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 IQ PIC, the byte count includes framing and control word overhead. (See Table 20 on page 252.)

Table 20: 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
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 Supported on Devices Running Junos OS” and “Juniper Networks Enterprise-Specific SNMP Traps” in the [Junos OS SNMP MIBs and Traps Reference](#).

- Related Documentation**
- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235
 - Understanding RMON for Monitoring Service Quality on page 237
 - Defining and Measuring Network Availability on page 241
 - Measuring Performance on page 253

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 21 on page 253).

Table 21: Performance Metrics

Metric:	Average delay
Description	Average round-trip time (in milliseconds) between two measurement points.
MIB name	DISMAN-PING-MIB (RFC 2925)
Variable name	pingResultsAverageRtt
Variable OID	pingResultsEntry.6
Frequency (mins)	15 (or depending upon ping test frequency)
Allowable range	To be baselined
Managed objects	Each measured path in the network
Metric:	Interface utilization
Description	Utilization percentage of a logical connection.
MIB name	IF-MIB
Variable name	(ifInOctets & ifOutOctets) * 8 / ifSpeed
Variable OID	ifTable entries
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All operational interfaces in the network

Table 21: Performance Metrics (*continued*)

Metric:	Disk utilization
Description	Utilization of disk space within the Juniper Networks router
MIB name	HOST-RESOURCES-MIB (RFC 2790)
Variable name	hrStorageSize – hrStorageUsed
Variable OID	hrStorageEntry.5 – hrStorageEntry.6
Frequency (mins)	1440
Allowable range	To be baselined
Managed objects	All Routing Engine hard disks
Metric:	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
Metric:	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
Metric:	LSP utilization

Table 21: Performance Metrics (*continued*)

Description	Utilization of the MPLS label-switched path.
MIB name	MPLS-MIB
Variable name	mplsPathBandwidth / (mplsLspOctets * 8)
Variable OID	mplsLspEntry.21 and mplsLspEntry.3
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	All label-switched paths in the network
Metric:	Output queue size
Description	Size, in packets, of each output queue per forwarding class, per interface.
MIB name	JUNIPER-COS-MIB
Variable name	jnxCosIfqQedPkts
Variable OID	jnxCosIfqStatsEntry.3
Frequency (mins)	60
Allowable range	To be baselined
Managed objects	For each forwarding class per interface in the network, once CoS is enabled.

This section includes the following topics:

- Measuring Class of Service on page 255
- Inbound Firewall Filter Counters per Class on page 256
- Monitoring Output Bytes per Queue on page 257
- Dropped Traffic on page 258

Measuring Class of Service

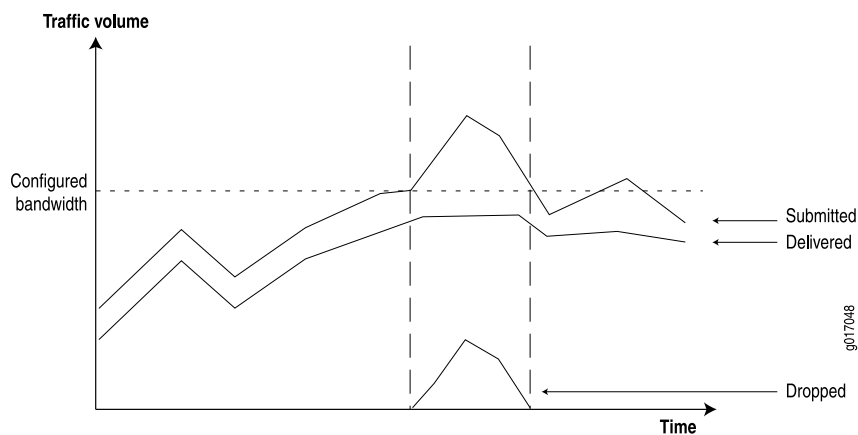
You can use class-of-service (CoS) mechanisms to regulate how certain classes of packets are handled within your network during times of peak congestion. Typically you must perform the following steps when implementing a CoS mechanism:

- Identify the type of packets that is 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 256.)

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 22 on page 257 shows additional filters used to match the other classes.

Table 22: Inbound Traffic Per Class

DSCP Value	Firewall Match Condition	Description
10	af11	Assured forwarding class 1 drop profile 1
12	af12	Assured forwarding class 1 drop profile 2
18	af21	Best effort class 2 drop profile 1
20	af22	Best effort class 2 drop profile 2
26	af31	Best effort class 3 drop profile 1

Any packet with a CoS DiffServ code point (DSCP) conforming to RFC 2474 can be counted in this way. The Juniper Networks enterprise-specific Firewall Filter MIB presents the counter information in the variables shown in Table 23 on page 257.

Table 23: 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 24 on page 258.)

Table 24: 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 25 on page 258.

Table 25: 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 26 on page 258.

Table 26: Dropped Traffic Counters

Indicator Name	Dropped Traffic
MIB	JUNIPER-COS-MIB
Table	jnxCosIfqStatsTable

Table 26: Dropped Traffic Counters (*continued*)

Indicator Name	Dropped Traffic
Index	jnxCosIfqIfIndex.jnxCosIfqFc
Variables	jnxCosIfqTailDropPkts jnxCosIfqTotalRedDropPkts
Description	The number of tail-dropped or RED-dropped packets per interface per forwarding class
SNMP version	SNMPv2

Related Documentation

- Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 235
- Understanding RMON for Monitoring Service Quality on page 237
- Defining and Measuring Network Availability on page 241
- Measuring Health on page 246

PART 7

Accounting Options

- [Accounting Options Overview on page 263](#)
- [Configuring Accounting Options on page 265](#)
- [Summary of Accounting Options Configuration Statements on page 289](#)

CHAPTER 19

Accounting Options Overview

This chapter contains the following topic:

- Accounting Options Overview on page 263

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 27 on page 263.

Table 27: 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.

Related Documentation

- Understanding Device Management Functions in Junos OS on page 3
- Accounting Options Configuration on page 265
- Configuring Accounting-Data Log Files on page 268

- [Configuring the Interface Profile on page 271](#)
- [Configuring the Filter Profile on page 274](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 7](#)

CHAPTER 20

Configuring Accounting Options

This chapter contains the following topics:

- Accounting Options Configuration on page 265
- Configuring Accounting-Data Log Files on page 268
- Configuring the Interface Profile on page 271
- Configuring the Filter Profile on page 274
- Example: Configuring a Filter Profile on page 275
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 276
- Understanding Source Class Usage and Destination Class Usage Options on page 278
- Configuring SCU or DCU on page 278
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 281
- Configuring Class Usage Profiles on page 282
- Configuring the MIB Profile on page 284
- Configuring the Routing Engine Profile on page 286

Accounting Options Configuration

This topic contains the following sections:

- Accounting Options—Full Configuration on page 265
- Minimum Accounting Options Configuration on page 266

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;
    object-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;
    object-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 *logical-unit-number*]** hierarchy level. For more information about interface profiles, see the [Junos OS Network Interfaces Configuration Guide](#).

[edit interfaces]

```
interface-name {  
  accounting-profile profile-name;  
  unit logical-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.

Related Documentation

- Accounting Options Overview on page 263
- Understanding Device Management Functions in Junos OS on page 3
- Configuring Accounting-Data Log Files on page 268
- Configuring the Interface Profile on page 271
- Configuring the Filter Profile on page 274
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7

Configuring Accounting-Data Log 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 the name of the 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 269
- Configuring the Maximum Size of the File on page 269
- Configuring the Maximum Number of Files on page 270
- Configuring the Start Time for File Transfer on page 270
- Configuring the Transfer Interval of the File on page 270
- Configuring Archive Sites on page 271

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 back up 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;
```

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.

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 is represented 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.



TIP:

Junos OS saves the existing log file and creates a new file at the configured transfer-intervals irrespective of:

- Whether the file has reached the maximum size or not
- Whether an archive site is configured or not

When you have a relatively smaller transfer-interval configured and if no archive site is configured, there is a possibility of losing data as Junos OS overwrites the log files when the maximum number of log files is reached. To ensure that the log information is saved for a reasonably long time:

- Configure an archive site to archive the log files every time a new log file is created.
 - Configure the maximum value (2880 minutes) for transfer-interval so that new files are created less frequently; that is, only when the file exceeds the maximum size limit or once in 2 days.
-

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 about specifying valid FTP URLs, see the [Junos OS System Basics Configuration Guide](#). You can specify more than one URL, in any order. When a file is archived, the router or switch 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**.

Related Documentation

- Accounting Options Overview on page 263
- Understanding Device Management Functions in Junos OS on page 3
- Accounting Options Configuration on page 265
- Configuring the Interface Profile on page 271
- Configuring the Filter Profile on page 274
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7

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 *logical-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 OS Routing Policy Configuration Guide](#).

To configure an interface profile, you perform the tasks described in the following sections:

- Configuring Fields on page 272
- Configuring the File Information on page 272
- Configuring the Interval on page 272
- Example: Configuring the Interface Profile on page 272

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

```

Related Documentation

- Accounting Options Overview on page 263
- Understanding Device Management Functions in Junos OS on page 3

- Accounting Options Configuration on page 265
- Configuring Accounting-Data Log Files on page 268
- Configuring the Filter Profile on page 274
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7

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 about firewall filters, see the [Junos OS Network Interfaces Configuration Guide](#).

To configure a filter profile, perform the tasks described in the following sections:

- Configuring the Counters on page 274
- Configuring the File Information on page 274
- Configuring the Interval on page 275

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, Junos OS closes the file and starts a new one. By default, the transfer interval value is 30 minutes. If the transfer interval is not configured, Junos OS 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.

Related Documentation

- Accounting Options Overview on page 263
- Understanding Device Management Functions in Junos OS on page 3
- Accounting Options Configuration on page 265
- Configuring Accounting-Data Log Files on page 268
- Configuring the Interface Profile on page 271
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7
- Example: Configuring a Filter Profile on page 275
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 276

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
```

- Related Documentation**
- [Configuring the Filter Profile on page 274](#)
 - [Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 276](#)

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
```

Related Documentation

- Configuring the Filter Profile on page 274
- Configuring the Interface Profile on page 271

Understanding Source Class Usage and Destination Class Usage Options

You can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as source classes and *destination classes*. You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces.

Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

On T Series and M320 routers, 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 routers, SCU and DCU accounting is performed before the packet enters the fabric.
- On T Series and M320 routers, 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 routers, 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 OS Routing Policy Configuration Guide](#), the [Junos OS Network Interfaces Configuration Guide](#), and the [Junos OS Feature Guides](#).

Related Documentation

- Configuring SCU or DCU on page 278
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 281
- Configuring Class Usage Profiles on page 282
- Configuring the MIB Profile on page 284
- Configuring the Routing Engine Profile on page 286

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 279
- Applying the Policy to the Forwarding Table on page 279
- Enabling Accounting on Inbound and Outbound Interfaces on page 279

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 about configuring route filters and source classes in a routing policy, see the [Junos OS Routing Policy Configuration Guide](#) and the [Junos OS Network Interfaces Configuration Guide](#).

**Related
Documentation**

- Understanding Source Class Usage and Destination Class Usage Options on page 278
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 281
- Configuring Class Usage Profiles on page 282
- Configuring the MIB Profile on page 284
- Configuring the Routing Engine Profile on page 286

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 281
- Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface on page 281
- Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface on page 282

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 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 OS Network Interfaces Configuration Guide](#).

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 278
- Configuring SCU or DCU on page 278
- Configuring Class Usage Profiles on page 282
- Configuring the MIB Profile on page 284
- Configuring the Routing Engine Profile on page 286

Configuring Class Usage Profiles

To collect class usage statistics, perform the tasks described in these sections:

- Configuring a Class Usage Profile on page 282
- Configuring the File Information on page 283
- Configuring the Interval on page 283
- Creating a Class Usage Profile to Collect Source Class Usage Statistics on page 283
- Creating a Class Usage Profile to Collect Destination Class Usage Statistics on page 284

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 specify 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
```

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 278
- Configuring SCU or DCU on page 278
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 281
- Configuring the Routing Engine Profile on page 286

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;
  object-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 285
- Configuring the Interval on page 285
- Configuring the MIB Operation on page 285
- Configuring MIB Object Names on page 286
- Example: Configuring a MIB Profile on page 286

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]  
object-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;  
  }  
}
```

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 278
- Configuring SCU or DCU on page 278
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 281
- Configuring Class Usage Profiles on page 282
- Configuring the Routing Engine Profile on page 286

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 287
- Configuring the File Information on page 287
- Configuring the Interval on page 287
- Example: Configuring a Routing Engine Profile on page 287

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;  
  }  
}
```

**Related
Documentation**

- [Understanding Source Class Usage and Destination Class Usage Options on page 278](#)
- [Configuring SCU or DCU on page 278](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 281](#)
- [Configuring Class Usage Profiles on page 282](#)
- [Configuring the MIB Profile on page 284](#)

CHAPTER 21

Summary of Accounting Options Configuration Statements

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

accounting-options

Syntax	accounting-options {...} }
Hierarchy Level	[edit]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Configure options for accounting statistics collection.
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuration Statements at the [edit accounting-options] Hierarchy Level on page 7• Accounting Options Configuration on page 265

archive-sites

Syntax	<pre>archive-sites { site-name; }</pre>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure an archive site. If more than one site name is configured, an ordered list of archive sites for the accounting-data log files is created. When a file is archived, the router or switch attempts to transfer the file to the first URL in the list, moving to the next site only if the transfer does not succeed. The log file is stored at the archive site with a filename of the format <i>router-name_log-filename_timestamp</i> .
Options	<i>site-name</i> —Any valid FTP URL to a destination. For information about specifying valid FTP URLs, see the Junos System Basics Configuration Guide .
Required Privilege Level	snmp—To view this statement in the configuration. snmp-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring Archive Sites on page 271

class-usage-profile

Syntax	<pre>class-usage-profile <i>profile-name</i> { file <i>filename</i>; interval <i>minutes</i>; source-classes { <i>source-class-name</i>; } destination-classes { <i>destination-class-name</i>; } }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Create a class usage profile, which is used to log class usage statistics to a file in the <code>/var/log</code> directory. The class usage profile logs class usage statistics for the configured source classes on every interface that has destination-class-usage configured.</p> <p>For information about configuring source classes, see the Junos Routing Protocols Configuration Guide. For information about configuring source class usage, see the Junos Network Interfaces Configuration Guide.</p>
Options	<p>profile-name—Name of the destination class profile.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring Class Usage Profiles on page 282

counters

Syntax	<pre>counters { counter-name; }</pre>
Hierarchy Level	[edit accounting-options filter-profile <i>profile-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Names of counters for which filter profile statistics are collected. The packet and byte counts for the counters are logged to a file in the <i>/var/log</i> directory.
Options	<i>counter-name</i> —Name of the counter.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Counters on page 274

destination-classes

Syntax	<pre>destination-classes { destination-class-name; }</pre>
Hierarchy Level	[edit accounting-options class-usage-profile <i>profile-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the destination classes for which statistics are collected.
Options	<i>destination-class-name</i> —Name of the destination class to include in the source class usage profile.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring a Class Usage Profile on page 282

fields

See the following sections:

- **fields (for Interface Profiles) on page 293**
- **fields (for Routing Engine Profiles) on page 294**

fields (for Interface Profiles)

Syntax	<pre>fields { field-name; }</pre>
Hierarchy Level	[edit accounting-options interface-profile <i>profile-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Statistics to collect in an accounting-data log file for an interface.
Options	<p><i>field-name</i>—Name of the field:</p> <ul style="list-style-type: none"> • 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
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring the Interface Profile on page 271

fields (for Routing Engine Profiles)

Syntax	<pre>fields { <i>field-name</i>; }</pre>
Hierarchy Level	[edit accounting-options routing-engine-profile <i>profile-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Statistics to collect in an accounting-data log file for a Routing Engine.
Options	<p><i>field-name</i>—Name of the field:</p> <ul style="list-style-type: none">• cpu-load-1—Average system load over the last 1 minute• cpu-load-5—Average system load over the last 5 minutes• cpu-load-15—Average system load over the last 15 minutes• date—Date, in YYYYMMDD format• host-name—Hostname for the router• time-of-day—Time of day, in HHMMSS format• uptime—Time since last reboot, in seconds
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring the Routing Engine Profile on page 286

file

See the following sections:

- [file \(Associating with a Profile\)](#) on page 295
- [file \(Configuring a Log File\)](#) on page 296

file (Associating with a Profile)

Syntax	<code>file <i>filename</i>;</code>
Hierarchy Level	[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>]
Release Information	Statement introduced before Junos OS Release 7.4. The [edit accounting-options mib-profile <i>profile-name</i>] hierarchy added in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series Switches.
Description	Specify the accounting log file associated with the profile.
Options	<i>filename</i> —Name of the log file. You must specify a filename already configured in the file statement at the [edit accounting-options] hierarchy level.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Configuring the Interface Profile on page 271 • Configuring the Filter Profile on page 274 • Configuring the MIB Profile on page 284 • Configuring the Routing Engine Profile on page 286

file (Configuring a Log File)

Syntax	<pre>file <i>filename</i> { archive-sites { <i>site-name</i>; } files <i>number</i>; nonpersistent; size <i>bytes</i>; source-classes <i>time</i>; transfer-interval <i>minutes</i>; }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify a log file to be used for accounting data.
Options	<i>filename</i> —Name of the file in which to write accounting data. The remaining statements are explained separately.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring Accounting-Data Log Files on page 268

files

Syntax	<pre>files <i>number</i>;</pre>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the maximum number of log files to be used for accounting data.
Options	<i>number</i> —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 <i>number</i> is 3 and the default value is 10.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring Accounting-Data Log Files on page 268

filter-profile

Syntax	<pre>filter-profile <i>profile-name</i> { counters { <i>counter-name</i>; } file <i>filename</i>; interval <i>minutes</i>; }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Create a profile to filter and collect packet and byte count statistics and write them to a file in the <code>/var/log</code> directory. To apply the profile to a firewall filter, you include the accounting-profile statement at the [edit firewall filter <i>filter-name</i>] hierarchy level. For more information about firewall filters, see the Junos Network Interfaces Configuration Guide.</p>
Options	<p><i>profile-name</i>—Name of the filter profile.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring the Filter Profile on page 274

interface-profile

Syntax	<pre>interface-profile <i>profile-name</i> { fields { <i>field-name</i>; } file <i>filename</i>; interval <i>minutes</i>; }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Create a profile to filter and collect error and packet statistics and write them to a file in the <code>/var/log</code> directory. You can specify an interface profile for either a physical or a logical interface.
Options	<p><i>profile-name</i>—Name of the interface profile.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Interface Profile on page 271


interval

Syntax	<code>interval <i>minutes</i>;</code>
Hierarchy Level	[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>]
Release Information	Statement introduced before Junos OS Release 7.4. The [edit accounting-options mib-profile <i>profile-name</i>] hierarchy level added in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify how often statistics are collected for the accounting profile.
Options	<i>minutes</i> —Length of time between each collection of statistics. Range: 1 through 2880 minutes Default: 30 minutes
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Interface Profile on page 271 Configuring the Filter Profile on page 274 Configuring the MIB Profile on page 284 Configuring the Routing Engine Profile on page 286

mib-profile

Syntax	<pre>mib-profile <i>profile-name</i> { file <i>filename</i>; interval <i>minutes</i>; object-names { <i>mib-object-name</i>; } operation <i>operation-name</i>; }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	Statement introduced in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Create a MIB profile to collect selected MIB statistics and write them to a file in the <code>/var/log</code> directory.
Options	<p><i>profile-name</i>—Name of the MIB statistics profile.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the MIB Profile on page 284

nonpersistent

Syntax	<code>nonpersistent;</code>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced in Junos OS Release 8.3.
Description	For J Series Services Routers only. Store 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.
	<div>  <p>NOTE: If log files for accounting data are stored on DRAM, these files are lost when you reboot the router. Therefore, you should back up these files periodically.</p> </div>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Storage Location of the File on page 269

object-names

Syntax	<pre>object-names { mib-object-name; }</pre>
Hierarchy Level	[edit accounting-options mib-profile <i>profile-name</i>]
Release Information	Statement introduced in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the name of each MIB object for which MIB statistics are collected for an accounting-data log file.
Options	mib-object-name —Name of a MIB object. You can specify more than one MIB object name.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the MIB Profile on page 284

operation

Syntax	<code>operation operation-name;</code>
Hierarchy Level	[edit accounting-options mib-profile <i>profile-name</i>]
Release Information	Statement introduced in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the name of the operation used to collect MIB statistics for an accounting-data log file.
Options	operation-name —Name of the operation to use. You can specify a get , get-next , or walk operation. Default: walk
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the MIB Profile on page 284

routing-engine-profile

Syntax	<pre>routing-engine-profile <i>profile-name</i> { fields { <i>field-name</i>; } file <i>filename</i>; interval <i>minutes</i>; }</pre>
Hierarchy Level	[edit accounting-options]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Create a Routing Engine profile to collect selected Routing Engine statistics and write them to a file in the <code>/var/log</code> directory.
Options	profile-name —Name of the Routing Engine statistics profile. The remaining statements are explained separately.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Routing Engine Profile on page 286

size

Syntax	<code>size bytes;</code>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify attributes of an accounting-data log file.
Options	<p>bytes—Maximum size of each log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. If you do not specify a size, the file is closed, archived, and renamed when the time specified for the transfer interval is exceeded.</p> <p>Syntax: <i>x</i> to specify bytes, <i>xk</i> to specify KB, <i>xm</i> to specify MB, <i>xg</i> to specify GB</p> <p>Range: 256 KB through 1 GB</p>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring the Maximum Size of the File on page 269

source-classes

Syntax	<pre>source-classes { source-class-name; }</pre>
Hierarchy Level	[edit accounting-options class-usage-profile <i>profile-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the source classes for which statistics are collected.
Options	source-class-name —Name of the source class to include in the class usage profile.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Configuring a Class Usage Profile on page 282

start-time

Syntax	<code>start-time <i>time</i>;</code>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced in Junos OS Release 8.2. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the start time for transfer of an accounting-data log file.
Options	<i>time</i> —Start time for file transfer. Syntax: <i>YYYY-MM-DD.hh:mm</i>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Start Time for File Transfer on page 270

transfer-interval

Syntax	<code>transfer-interval <i>minutes</i>;</code>
Hierarchy Level	[edit accounting-options file <i>filename</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Specify the length of time the file remains open and receives new statistics before it is closed and transferred to an archive site.
Options	<i>minutes</i> —Time the file remains open and receives new statistics before it is closed and transferred to an archive site. Range: 5 through 2880 minutes Default: 30 minutes
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Configuring the Transfer Interval of the File on page 270

PART 8

Index

- Index on page 307
- Index of Statements and Commands on page 313

Index

Symbols

#, comments in configuration statements.....	xxviii
(), in syntax descriptions.....	xxviii
/var/log/mib2d file.....	42
/var/log/snmpd file.....	42
< >, in syntax descriptions.....	xxvii
[], in configuration statements.....	xxviii
{ }, in configuration statements.....	xxviii
(pipe), in syntax descriptions.....	xxviii

A

access statement	
usage guidelines.....	58
access-list statement.....	127
accounting options	
configuration.....	265
overview.....	263
accounting profiles	
filter.....	274
interface.....	271
MIB.....	284
Routing Engine.....	286
accounting-options statement.....	289
address statement	
SNMPv3.....	151
usage guidelines.....	69
address-mask statement.....	152
usage guidelines.....	69
agent, SNMP.....	22
agent-address statement.....	128
alarm statement	
RMON.....	213
usage guidelines.....	197
archive-sites statement	
accounting.....	290
usage guidelines.....	271
authentication-md5 statement.....	152
usage guidelines.....	55
authentication-none statement.....	153
usage guidelines.....	56

authentication-password statement.....	154
usage guidelines.....	55
authentication-sha statement.....	155
usage guidelines.....	55
authorization statement.....	128
usage guidelines.....	29

B

braces, in configuration statements.....	xxviii
brackets	
angle, in syntax descriptions.....	xxvii
square, in configuration statements.....	xxviii

C

categories statement.....	129
usage guidelines.....	36
Class 1 MIB objects.....	116
Class 2 MIB objects.....	120
Class 3 MIB objects.....	121
Class 4 MIB objects.....	122
class-usage-profile statement.....	291
usage guidelines.....	282
client list	
adding to SNMP community.....	30
client-list statement.....	129
usage guidelines.....	30
client-list-name statement.....	130
usage guidelines.....	30
clients statement.....	130
usage guidelines.....	29
comments, in configuration statements.....	xxviii
commit-delay statement.....	131
usage guidelines.....	28
community statement	
RMON.....	214
usage guidelines.....	200
SNMP.....	132
usage guidelines.....	29
community string, SNMP.....	29
community-name statement.....	156
usage guidelines.....	80

contact statement.....	133
usage guidelines.....	26
conventions	
text and syntax.....	xxvii
CoS	
measuring.....	255
counters statement.....	292
curly braces, in configuration statements.....	xxviii
customer support.....	xxviii
contacting JTAC.....	xxviii

D

description statement	
RMON.....	214
usage guidelines (alarms).....	197
usage guidelines (events).....	200
SNMP.....	133
usage guidelines.....	27
destination-classes statement.....	292
usage guidelines.....	282
destination-port statement	
SNMP.....	134
usage guidelines.....	36
documentation	
comments on.....	xxviii
dropped traffic	
measuring.....	258

E

engine-id statement	
SNMPv3.....	157
usage guidelines.....	53
enterprise-oid statement.....	134
enterprise-specific MIBs, listed.....	123
event statement.....	215
usage guidelines.....	200

F

falling-event-index statement.....	215
usage guidelines.....	197
falling-threshold statement	
health monitor.....	229
usage guidelines.....	227
RMON.....	216
falling-threshold-interval statement	
RMON.....	217
usage guidelines.....	198

fields statement	
for interface profiles.....	293
usage guidelines.....	272
for Routing Engine profiles.....	294
usage guidelines.....	287
file statement	
accounting (associating with profile).....	295
usage guidelines (filter profile).....	274
usage guidelines (interface profile).....	272
usage guidelines (MIB profile).....	285
usage guidelines (Routing Engine profile).....	287
accounting (configuring log file).....	296
usage guidelines.....	268
files statement.....	296
filter profile.....	274
filter-duplicates statement.....	135
usage guidelines.....	27
filter-interfaces statement.....	135
filter-profile statement.....	297
usage guidelines.....	274
filtering get SNMP requests.....	27
font conventions.....	xxvii

G

Get requests, SNMP.....	19
group statement	
SNMPv3 (for access privileges).....	159
usage guidelines.....	64
SNMPv3 (for configuring).....	158
usage guidelines.....	59

H

health metrics of network.....	246
health-monitor statement.....	230
usage guidelines.....	227

I

icons defined, notice.....	xxvi
ILMI.....	15
informs SNMP See SNMP informs	
integrated local management interface See ILMI	
interface profile.....	271
interface statement	
SNMP.....	136
usage guidelines.....	39
interface-profile statement.....	298
usage guidelines.....	271
interfaces limiting SNMP access.....	39

interval statement	
accounting.....	299
usage guidelines (filter profile).....	275
usage guidelines (interface profile).....	272
usage guidelines (MIB profile).....	285
usage guidelines (Routing Engine profile).....	287
health monitor.....	230
usage guidelines.....	227
RMON.....	217
usage guidelines.....	198
IPv6 SNMP community string.....	29
J	
jnxRmonAlarmTable.....	204
Juniper Networks MIB objects.....	112
K	
key performance indicators.....	236
L	
local-engine statement.....	161
location statement	
SNMP.....	136
usage guidelines.....	26
logical-system statement.....	137
logical-system-trap-filter statement.....	138
M	
Management Information Base See MIBs	
manuals	
comments on.....	xxviii
master agent, SNMP.....	22
measurement tests	
proxy ping.....	244
message-processing-model statement.....	162
usage guidelines.....	73
MIB object classes.....	106
MIB profile.....	284
mib-profile statement.....	300
usage guidelines.....	284
MIBs	
enterprise-specific, listed.....	123
Ping	
use in ping test.....	88
view configuration example, SNMP.....	41
views	
SNMP.....	40
minimum accounting options configuration.....	266
monitoring	
service quality.....	235
N	
name statement.....	138
usage guidelines.....	28
network health	
measuring.....	246
network performance	
measuring.....	253
nonpersistent statement.....	301
accounting	
usage guidelines.....	269
nonvolatile statement.....	139
notice icons defined.....	xxvi
notify statement.....	163
usage guidelines.....	66
notify-filter statement	
for applying to target.....	164
usage guidelines.....	73
for configuring.....	164
usage guidelines.....	67
notify-view statement.....	165
usage guidelines.....	61
O	
object-names statement.....	301
objects-names statement	
for Routing Engine profiles	
usage guidelines.....	286
oid statement	
SNMP.....	139
usage guidelines.....	40
SNMPv3.....	165
usage guidelines.....	67
operation statement.....	302
for MIB profiles	
usage guidelines.....	285
P	
parameters statement.....	166
usage guidelines.....	72
parentheses, in syntax descriptions.....	xxviii
performance indicators.....	236
performance, monitoring.....	253
Ping MIB	
use in ping test.....	88
view configuration example	
SNMP.....	41

pingCtlTable.....	244
pingProbeHistoryTable.....	93
port statement	
SNMPv3.....	166
usage guidelines.....	70
prefix list	
adding to SNMP community.....	30
privacy-3des statement.....	167
usage guidelines.....	57
privacy-aes128 statement.....	168
usage guidelines.....	56
privacy-des statement.....	169
usage guidelines.....	57
privacy-none statement.....	169
usage guidelines.....	58
privacy-password statement.....	170
usage guidelines	
for 3DES algorithm.....	57
for AES algorithm.....	56
for DES algorithm.....	57
profiles, accounting	
filter.....	274
interface.....	271
MIB.....	284
Routing Engine.....	286
proxy ping	
measurement tests.....	244

R

read-view statement.....	171
usage guidelines.....	61
real-time performance monitoring	
in service provider networks.....	244
remote operations MIBs.....	87
remote-engine statement.....	172
request-type statement.....	218
RMON	
usage guidelines.....	199
retry-count statement.....	159
usage guidelines.....	77
rising-event-index statement.....	219
usage guidelines.....	197
rising-threshold statement	
health monitor.....	231
RMON.....	219
RMON alarm entries.....	196
RMON alarms.....	203, 239
RMON event entries.....	200
RMON events.....	208, 238

rmon statement.....	220
usage guidelines.....	238
Routing Engine profile.....	286
routing instances	
access lists	
configuring.....	111
SNMP	
enabling access.....	108
identifying.....	107
specifying.....	109
routing-engine-profile statement.....	302
usage guidelines.....	286
routing-instance statement	
SNMP.....	140
SNMPv3.....	173
usage guidelines.....	70
routing-instance-access.....	141

S

sample-type statement.....	220
usage guidelines	
for alarms.....	199
for events.....	200
security-level statement	
for access privileges.....	174
usage guidelines.....	60
for SNMP notifications.....	175
usage guidelines.....	74
security-model statement	
for access privileges.....	176
usage guidelines.....	60
for groups.....	177
usage guidelines.....	63
for SNMP notifications.....	177
usage guidelines.....	73
security-name statement.....	178
for community string.....	178
for security group.....	179
usage guidelines.....	64
for SNMP notifications.....	180
usage guidelines.....	74
security-to-group statement.....	181
usage guidelines.....	58
service quality	
monitoring.....	235
Set requests, SNMP.....	19
size statement	
accounting.....	303
usage guidelines.....	269

SNMP	
adding client lists and prefix lists.....	30
agent.....	19, 22
architecture.....	19
commit delay timer.....	28
community string.....	29
configuration	
version 3.....	49, 50
versions 1 and 2.....	24
filtering duplicate requests.....	27
limiting interface access.....	39
logging, enabling.....	88
manager.....	19
master agent.....	22
MIB views.....	40
remote operations.....	85
subagent.....	22
system contact.....	26
system description.....	27
system location.....	26, 136
system name.....	28
tracing operations.....	42
trap groups.....	36
trap notification for remote operations.....	87
trap options.....	33
views, setting.....	86
SNMP informs.....	75
snmp statement.....	141
usage guidelines	
SNMPv1 and SNMPv2.....	24
SNMPv3.....	49, 50
SNMP traps.....	20
source address configuration.....	33
system logging severity levels.....	21
snmp-community statement.....	181
SNMPv2	
Passive Monitoring Traps MIB.....	36
SNMPv3	
authentication, configuring.....	55
informs, configuring.....	75
local engine ID, configuring.....	53
minimum configuration.....	52
source-address statement.....	142
usage guidelines.....	33
source-classes statement.....	303
usage guidelines.....	282
start-time statement	
accounting.....	304
usage guidelines.....	270
startup-alarm statement.....	221
usage guidelines.....	199
subagent, SNMP.....	22
support, technical See technical support	
syntax conventions.....	xxvii
sysContact object, MIB II.....	26
sysDescription object, MIB II.....	27
sysLocation object, MIB II.....	26
syslog-subtag statement.....	221
usage guidelines.....	200
sysName object, MIB II.....	28
system contact, SNMP.....	26
system description, SNMP.....	27
system location, SNMP.....	26, 136
system logging severity levels, SNMP traps.....	21
system name, SNMP.....	28
T	
tag statement.....	182
SNMPv3	
usage guidelines.....	81
usage guidelines.....	66
tag-list statement.....	182
usage guidelines.....	70
target-address statement.....	183
usage guidelines.....	68
target-parameters statement.....	184
usage guidelines.....	72
targets statement.....	142
usage guidelines.....	36
technical support	
contacting JTAC.....	xxviii
timeout statement.....	160
usage guidelines.....	77
traceoptions statement.....	143
SNMP	
usage guidelines.....	42
Traceroute MIB.....	95
traceRouteHopsTable.....	100
tracing operations	
SNMP.....	42
transfer-interval statement	
accounting.....	304
usage guidelines.....	270
trap groups, SNMP.....	36
trap notification for SNMP remote operations.....	87
trap-group statement.....	145
usage guidelines.....	36

trap-options statement.....	146
usage guidelines.....	33
traps	
definition.....	20
type statement.....	222
usage guidelines.....	66

U

user statement	
SNMPv3.....	185
usm statement.....	186

V

v3 statement.....	188
usage guidelines.....	49, 50
vacm statement.....	190
usage guidelines.....	58
var/log/mib2d file.....	42
var/log/snmpd file.....	42
variable statement.....	222
usage guidelines.....	200
variable-length string indexes.....	87
version statement	
SNMP.....	147
usage guidelines.....	36
view statement	
SNMP (associating with community).....	148
usage guidelines.....	29
SNMP (configuring MIB view).....	149
usage guidelines.....	40
views, MIB	
SNMP.....	40, 86

W

write-view statement.....	191
usage guidelines.....	62

Index of Statements and Commands

A

access-list statement.....	127
accounting-options statement.....	289
address statement	
SNMPv3.....	151
address-mask statement.....	152
agent-address statement.....	128
alarm statement	
RMON.....	213
archive-sites statement	
accounting.....	290
authentication-md5 statement.....	152
authentication-none statement.....	153
authentication-password statement.....	154
authentication-sha statement.....	155
authorization statement.....	128

C

categories statement.....	129
class-usage-profile statement.....	291
client-list statement.....	129
client-list-name statement.....	130
clients statement.....	130
commit-delay statement.....	131
community statement	
RMON.....	214
SNMP.....	132
community-name statement.....	156
contact statement.....	133
counters statement.....	292

D

description statement	
RMON.....	214
SNMP.....	133
destination-classes statement.....	292
destination-port statement	
SNMP.....	134

E

engine-id statement	
SNMPv3.....	157
enterprise-oid statement.....	134
event statement.....	215

F

falling-event-index statement.....	215
falling-threshold statement	
health monitor.....	229
RMON.....	216
falling-threshold-interval statement	
RMON.....	217
fields statement	
for interface profiles.....	293
for Routing Engine profiles.....	294
file statement	
accounting (associating with profile).....	295
accounting (configuring log file).....	296
files statement.....	296
filter-duplicates statement.....	135
filter-interfaces statement.....	135
filter-profile statement.....	297

G

group statement	
SNMPv3 (for access privileges).....	159
SNMPv3 (for configuring).....	158

H

health-monitor statement.....	230
-------------------------------	-----

I

interface statement	
SNMP.....	136
interface-profile statement.....	298
interval statement	
accounting.....	299
health monitor.....	230
RMON.....	217

L

local-engine statement.....	161
location statement	
SNMP.....	136
logical-system statement.....	137
logical-system-trap-filter statement.....	138

M

message-processing-model statement.....	162
mib-profile statement.....	300

N

name statement.....	138
nonpersistent statement.....	301
nonvolatile statement.....	139
notify statement.....	163
notify-filter statement	
for applying to target.....	164
for configuring.....	164
notify-view statement.....	165

O

object-names statement.....	301
oid statement	
SNMP.....	139
SNMPv3.....	165
operation statement.....	302

P

parameters statement.....	166
port statement	
SNMPv3.....	166
privacy-3des statement.....	167
privacy-aes128 statement.....	168
privacy-des statement.....	169
privacy-none statement.....	169
privacy-password statement.....	170

R

read-view statement.....	171
remote-engine statement.....	172
request-type statement.....	218
retry-count statement.....	159
rising-event-index statement.....	219
rising-threshold statement	
health monitor.....	231
RMON.....	219
rmon statement.....	220
routing-engine-profile statement.....	302

routing-instance statement	
SNMP.....	140
SNMPv3.....	173
routing-instance-access.....	141

S

sample-type statement.....	220
security-level statement	
for access privileges.....	174
for SNMP notifications.....	175
security-model statement	
for access privileges.....	176
for groups.....	177
for SNMP notifications.....	177
security-name statement.....	178
for community string.....	178
for security group.....	179
for SNMP notifications.....	180
security-to-group statement.....	181
size statement	
accounting.....	303
snmp statement.....	141
snmp-community statement.....	181
source-address statement.....	142
source-classes statement.....	303
start-time statement	
accounting.....	304
startup-alarm statement.....	221
syslog-subtag statement.....	221

T

tag statement.....	182
tag-list statement.....	182
target-address statement.....	183
target-parameters statement.....	184
targets statement.....	142
timeout statement.....	160
traceoptions statement.....	143
transfer-interval statement	
accounting.....	304
trap-group statement.....	145
trap-options statement.....	146
type statement.....	222

U

user statement	
SNMPv3.....	185
usm statement.....	186

V

v3 statement.....	188
vacm statement.....	190
variable statement.....	222
version statement	
SNMP.....	147
view statement	
SNMP (associating with community).....	148
SNMP (configuring MIB view).....	149

W

write-view statement.....	191
---------------------------	-----

