



Junos[®] OS

Network Management Administration Guide for Routing Devices

Release

14.1



Published: 2014-09-27

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

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

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

Junos[®] OS Network Management Administration Guide for Routing Devices

14.1

Copyright © 2014, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xix
	Documentation and Release Notes	xix
	Supported Platforms	xix
	Using the Examples in This Manual	xix
	Merging a Full Example	xx
	Merging a Snippet	xx
	Documentation Conventions	xxi
	Documentation Feedback	xxiii
	Requesting Technical Support	xxiii
	Self-Help Online Tools and Resources	xxiii
	Opening a Case with JTAC	xxiv
Part 1	Overview	
Chapter 1	Network Management Overview	3
	Understanding Device Management Functions in Junos OS	3
	Understanding the Integrated Local Management Interface	6
Part 2	Network Monitoring Using SNMP	
Chapter 2	SNMP Overview	9
	Understanding the SNMP Implementation in Junos OS	9
	SNMP Architecture	9
	SNMP MIBs	10
	SNMP Traps and Informs	10
	Junos OS SNMP Agent Features	12
	SNMPv3 Overview	12
Chapter 3	SNMP MIBs and Traps Supported by Junos OS	15
	Standard SNMP MIBs Supported by Junos OS	15
	Juniper Networks Enterprise-Specific MIBs	34
	Juniper Networks Enterprise-Specific MIBs and Supported Devices	46
	SNMP MIB Objects Supported by Junos OS for the Set Operation	55
	Standard SNMP Traps Supported on Devices Running Junos OS	62
	Juniper Networks Enterprise-Specific SNMP Traps	63
Chapter 4	Loading MIB Files to a Network Management System	65
	Loading MIB Files to a Network Management System	65

Chapter 5	Configuring SNMP	69
	Configuration Statements at the [edit snmp] Hierarchy Level	69
	Configuring SNMP on a Device Running Junos OS	73
	Configuring the System Contact on a Device Running Junos OS	75
	Configuring the System Location for a Device Running Junos OS	76
	Configuring the System Description on a Device Running Junos OS	76
	Configuring SNMP Details	77
	Configuring the System Name	78
	Configuring the Commit Delay Timer	79
	Filtering Duplicate SNMP Requests	79
	Configuring SNMP Communities	80
	Examples: Configuring the SNMP Community String	83
	Adding a Group of Clients to an SNMP Community	83
	Configuring a Proxy SNMP Agent	85
	Configuring SNMP Traps	86
	Configuring SNMP Trap Options and Groups on a Device Running Junos OS	88
	Configuring SNMP Trap Options	88
	Configuring the Source Address for SNMP Traps	89
	Configuring the Agent Address for SNMP Traps	91
	Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps	91
	Configuring SNMP Trap Groups	92
	Example: Configuring SNMP Trap Groups	94
	Configuring the Interfaces on Which SNMP Requests Can Be Accepted	95
	Example: Configuring Secured Access List Checking	95
	Filtering Interface Information Out of SNMP Get and GetNext Output	96
	Configuring MIB Views	97
	Example: Ping Proxy MIB	98
Chapter 6	Configuring SNMPv3	101
	Complete SNMPv3 Configuration Statements	102
	Minimum SNMPv3 Configuration on a Device Running Junos OS	104
	Example: SNMPv3 Configuration	105
	Configuring the Local Engine ID	108
	Creating SNMPv3 Users	109
	Example: Creating SNMPv3 Users	110
	Configuring the SNMPv3 Authentication Type	111
	Configuring MD5 Authentication	111
	Configuring SHA Authentication	111
	Configuring No Authentication	112
	Configuring the SNMPv3 Encryption Type	112
	Configuring the Advanced Encryption Standard Algorithm	112
	Configuring the Data Encryption Algorithm	113
	Configuring Triple DES	113
	Configuring No Encryption	113
	Defining Access Privileges for an SNMP Group	114
	Configuring the Access Privileges Granted to a Group	115
	Configuring the Group	115
	Configuring the Security Model	116

Configuring the Security Level	116
Associating MIB Views with an SNMP User Group	116
Configuring the Notify View	117
Configuring the Read View	117
Configuring the Write View	118
Example: Configuring the Access Privileges Granted to a Group	118
Assigning Security Model and Security Name to a Group	119
Configuring the Security Model	119
Assigning Security Names to Groups	120
Configuring the Group	120
Example: Configuring a Security Group	121
Configuring SNMPv3 Traps on a Device Running Junos OS	121
Configuring the SNMPv3 Trap Notification	122
Example: Configuring SNMPv3 Trap Notification	123
Configuring the Trap Notification Filter	124
Configuring the Trap Target Address	124
Configuring the Address	125
Configuring the Address Mask	125
Configuring the Port	126
Configuring the Routing Instance	126
Configuring the Trap Target Address	126
Applying Target Parameters	127
Example: Configuring the Tag List	127
Defining and Configuring the Trap Target Parameters	128
Applying the Trap Notification Filter	129
Configuring the Target Parameters	129
Configuring the Message Processing Model	129
Configuring the Security Model	130
Configuring the Security Level	130
Configuring the Security Name	130
Configuring SNMP Informs	131
Configuring the Remote Engine and Remote User	132
Example: Configuring the Remote Engine ID and Remote Users	133
Configuring the Inform Notification Type and Target Address	136
Example: Configuring the Inform Notification Type and Target Address	138
Configuring the SNMPv3 Community	138
Configuring the Community Name	139
Configuring the Context	139
Configuring the Security Names	140
Configuring the Tag	140
Example: Configuring an SNMPv3 Community	140
Chapter 7	
Configuring SNMP for Routing Instances	143
Understanding SNMP Support for Routing Instances	143
SNMP MIBs Supported for Routing Instances	144
Support Classes for MIB Objects	154
SNMP Traps Supported for Routing Instances	155

	Identifying a Routing Instance	156
	Enabling SNMP Access over Routing Instances	157
	Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community	157
	Example: Configuring Interface Settings for a Routing Instance	158
	Configuring Access Lists for SNMP Access over Routing Instances	159
Chapter 8	Configuring SNMP Remote Operations	161
	SNMP Remote Operations Overview	161
	SNMP Remote Operation Requirements	162
	Setting SNMP Views	162
	Example: Setting SNMP Views	162
	Setting Trap Notification for Remote Operations	163
	Example: Setting Trap Notification for Remote Operations	163
	Using Variable-Length String Indexes	163
	Example: Set Variable-Length String Indexes	163
	Enabling Logging	164
	Using the Ping MIB for SNMP Remote Operations	164
	Starting a Ping Test	164
	Using Multiple Set Protocol Data Units (PDUs)	165
	Using a Single Set PDU	165
	Monitoring a Running Ping Test	166
	pingResultsTable	166
	pingProbeHistoryTable	167
	Generating Traps	168
	Gathering Ping Test Results	169
	Stopping a Ping Test	170
	Interpreting Ping Variables	170
	Using the Traceroute MIB for SNMP Remote Operations	171
	Starting a Traceroute Test	172
	Using Multiple Set PDUs	172
	Using a Single Set PDU	172
	Monitoring a Running Traceroute Test	173
	traceRouteResultsTable	173
	traceRouteProbeResultsTable	174
	traceRouteHopsTable	175
	Generating Traps	176
	Monitoring Traceroute Test Completion	177
	Gathering Traceroute Test Results	178
	Stopping a Traceroute Test	179
	Interpreting Traceroute Variables	180
Chapter 9	Tracing SNMP Activity	181
	Tracing SNMP Activity on a Device Running Junos OS	181
	Configuring the Number and Size of SNMP Log Files	182
	Configuring Access to the Log File	182
	Configuring a Regular Expression for Lines to Be Logged	183
	Configuring the Trace Operations	183
	Example: Tracing SNMP Activity	184

Chapter 10	SNMP FAQs	187
	Junos OS SNMP FAQs Overview	187
	Junos OS SNMP FAQs	188
	Junos OS SNMP Support FAQs	188
	Junos OS MIBs FAQs	189
	Junos OS SNMP Configuration FAQs	196
	SNMPv3 FAQs	200
	SNMP Interaction with Juniper Networks Devices FAQs	202
	SNMP Traps and Informs FAQs	204
	Junos OS Dual Routing Engine Configuration FAQs	210
	SNMP Support for Routing Instances FAQs	211
	SNMP Counters FAQs	212
Part 3	Remote Monitoring (RMON) with SNMP	
Chapter 11	RMON Overview	217
	Understanding RMON Alarms	217
	alarmTable	217
	jnxRmonAlarmTable	218
	Understanding RMON Events	219
	eventTable	219
Chapter 12	Configuring RMON Alarms and Events	221
	Understanding RMON Alarms and Events Configuration	221
	Minimum RMON Alarm and Event Entry Configuration	222
	Configuring an RMON Alarm Entry and Its Attributes	222
	Configuring the Alarm Entry	223
	Configuring the Description	223
	Configuring the Falling Event Index or Rising Event Index	223
	Configuring the Falling Threshold or Rising Threshold	224
	Configuring the Interval	224
	Configuring the Falling Threshold Interval	224
	Configuring the Request Type	225
	Configuring the Sample Type	225
	Configuring the Startup Alarm	225
	Configuring the System Log Tag	226
	Configuring the Variable	226
	Configuring an RMON Event Entry and Its Attributes	226
	Example: Configuring an RMON Alarm and Event Entry	227
Chapter 13	Monitoring RMON Alarms and Events	229
	Using alarmTable to Monitor MIB Objects	229
	Creating an Alarm Entry	229
	Configuring the Alarm MIB Objects	229
	alarmInterval	230
	alarmVariable	230
	alarmSampleType	230
	alarmValue	230
	alarmStartupAlarm	230
	alarmRisingThreshold	231

	alarmFallingThreshold	231
	alarmOwner	231
	alarmRisingEventIndex	231
	alarmFallingEventIndex	231
	Activating a New Row in alarmTable	232
	Modifying an Active Row in alarmTable	232
	Deactivating a Row in alarmTable	232
	Using eventTable to Log Alarms	232
	Creating an Event Entry	232
	Configuring the MIB Objects	233
	eventType	233
	eventCommunity	233
	eventOwner	233
	eventDescription	234
	Activating a New Row in eventTable	234
	Deactivating a Row in eventTable	234
Chapter 14	Using RMON to Monitor Network Service Quality	235
	Understanding RMON for Monitoring Service Quality	235
	Setting Thresholds	235
	RMON Command-Line Interface	236
	RMON Event Table	237
	RMON Alarm Table	237
	Troubleshooting RMON	238
	Understanding Measurement Points, Key Performance Indicators, and Baseline Values	239
	Measurement Points	239
	Basic Key Performance Indicators	240
	Setting Baselines	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 4	Health Monitoring with SNMP	
Chapter 15	Configuring Health Monitoring	263
	Configuring Health Monitoring on Devices Running Junos OS	263
	Monitored Objects	264
	Minimum Health Monitoring Configuration	265
	Configuring the Falling Threshold or Rising Threshold	265
	Configuring the Interval	265

	Log Entries and Traps	266
	Example: Configuring Health Monitoring	266
Part 5	Gathering Statistics for Accounting Purposes Using Accounting Options, Source Class Usage and Destination Class Usage Options	
Chapter 16	Accounting Options, Source Class Usage and Destination Class Usage Options Overview	269
	Accounting Options Overview	269
	Understanding Source Class Usage and Destination Class Usage Options	270
Chapter 17	Configuring Accounting Options, Source Class Usage and Destination Class Usage Options	273
	Configuration Statements at the [edit accounting-options] Hierarchy Level	273
	Accounting Options Configuration	274
	Accounting Options—Full Configuration	275
	Minimum Accounting Options Configuration	276
	Configuring Accounting-Data Log Files	278
	Configuring the Storage Location of the File	278
	Configuring the Maximum Size of the File	279
	Configuring the Maximum Number of Files	279
	Configuring the Start Time for File Transfer	279
	Configuring the Transfer Interval of the File	279
	Configuring Archive Sites	280
	Configuring the Interface Profile	281
	Configuring Fields	281
	Configuring the File Information	281
	Configuring the Interval	282
	Example: Configuring the Interface Profile	282
	Configuring the Filter Profile	283
	Configuring the Counters	284
	Configuring the File Information	284
	Configuring the Interval	285
	Example: Configuring a Filter Profile	285
	Example: Configuring Interface-Specific Firewall Counters and Filter Profiles	286
	Configuring SCU or DCU	287
	Creating Prefix Route Filters in a Policy Statement	288
	Applying the Policy to the Forwarding Table	288
	Enabling Accounting on Inbound and Outbound Interfaces	288
	Configuring SCU on a Virtual Loopback Tunnel Interface	290
	Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC	290
	Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface	290
	Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface	291
	Configuring Class Usage Profiles	291
	Configuring a Class Usage Profile	291
	Configuring the File Information	292

	Configuring the Interval	292
	Creating a Class Usage Profile to Collect Source Class Usage Statistics . . .	292
	Creating a Class Usage Profile to Collect Destination Class Usage Statistics	293
	Configuring the MIB Profile	293
	Configuring the File Information	294
	Configuring the Interval	294
	Configuring the MIB Operation	294
	Configuring MIB Object Names	295
	Example: Configuring a MIB Profile	295
	Configuring the Routing Engine Profile	295
	Configuring Fields	296
	Configuring the File Information	296
	Configuring the Interval	296
	Example: Configuring a Routing Engine Profile	296
Part 6	Configuration Statements and Operational Commands	
Chapter 18	Configuration Statements: Accounting Options, Source Class Usage and Destination Class Usage Options	301
	accounting-options	302
	archive-sites	302
	class-usage-profile	303
	counters	304
	destination-classes	304
	fields (for Interface Profiles)	305
	fields (for Routing Engine Profiles)	306
	file (Associating with a Profile)	307
	file (Configuring a Log File)	308
	files	309
	filter-profile	310
	interface-profile	311
	interval (Accounting Options)	312
	mib-profile	313
	nonpersistent	314
	object-names	314
	operation	315
	routing-engine-profile	315
	size	316
	source-classes	316
	start-time (Log File Transfer)	317
	transfer-interval	317
Chapter 19	Configuration Statements: Health Monitoring	319
	falling-threshold	319
	health-monitor	320
	interval (SNMP Health Monitor)	320
	rising-threshold (SNMP Health Monitor)	321

Chapter 20	Configuration Statements: Remote Monitoring (RMON)	323
	alarm (SNMP RMON)	324
	community (SNMP RMON)	325
	description (SNMP RMON)	325
	event (SNMP)	326
	falling-event-index	326
	falling-threshold	327
	falling-threshold-interval	328
	interval (SNMP RMON)	328
	request-type	329
	rising-event-index	330
	rising-threshold (SNMP RMON)	330
	rmon	331
	sample-type	331
	startup-alarm	332
	syslog-subtag	332
	type (SNMP RMON)	333
	variable	333
Chapter 21	Configuration Statements: SNMP	335
	access-list	336
	agent-address	337
	alarm-id	338
	alarm-list-name	339
	alarm-management	340
	alarm-state	341
	authorization	342
	categories	342
	client-list	343
	client-list-name	343
	clients	344
	commit-delay	344
	community (SNMP)	345
	contact (SNMP)	346
	description (SNMP)	347
	destination-port	348
	enterprise-oid	348
	filter-duplicates	349
	filter-interfaces	350
	interface (SNMP)	351
	location (SNMP)	351
	logical-system (SNMP)	352
	logical-system-trap-filter	353
	name	353
	nonvolatile	354
	oid	354
	proxy (snmp)	355
	routing-instance (SNMP)	356
	routing-instance-access	357

	snmp	357
	source-address (SNMP)	358
	targets	358
	traceoptions (SNMP)	359
	trap-group	361
	trap-options	362
	version (SNMP)	363
	view (Associating a MIB View with a Community)	363
	view (Configuring a MIB View)	364
Chapter 22	Configuration Statements: SNMPv3	365
	address (SNMP)	366
	address-mask	367
	authentication-md5	367
	authentication-none	368
	authentication-password	369
	authentication-sha	370
	community-name (SNMP)	371
	context (SNMPv3)	372
	engine-id (SNMP)	373
	group (Configuring Access Privileges)	374
	group (Associating a Security Name)	375
	retry-count (SNMPv3)	375
	timeout (SNMP)	376
	local-engine	377
	message-processing-model	378
	notify	379
	notify-filter (Applying to the Management Target)	380
	notify-filter (Configuring the Profile Name)	380
	notify-view	381
	oid (SNMP)	381
	parameters	382
	port (SNMP)	382
	privacy-3des	383
	privacy-aes128	384
	privacy-des	385
	privacy-none	385
	privacy-password	386
	read-view	387
	remote-engine	388
	routing-instance (SNMPv3)	389
	security-level (Defining Access Privileges)	390
	security-level (Generating SNMP Notifications)	391
	security-model (Access Privileges)	392
	security-model (Group)	393
	security-model (SNMP Notifications)	394
	security-name (Community String)	395
	security-name (Security Group)	396
	security-name (SNMP Notifications)	397

	security-to-group	398
	snmp-community	399
	tag (SNMPv3)	399
	tag-list	400
	target-address	401
	target-parameters	402
	type (SNMPv3)	403
	user	403
	usm	404
	v3	406
	vacm	408
	write-view	409
Chapter 23	Operational Commands	411
	clear ilmi statistics	412
	clear snmp history	413
	clear snmp statistics	414
	request snmp spoof-trap	416
	show interfaces snmp-index	422
	show ilmi statistics	423
	show snmp health-monitor	426
	show snmp inform-statistics	433
	show snmp mib	435
	show snmp rmon	438
	show snmp statistics	442
	show snmp v3	446
Part 7	Index	
	Index	451

List of Figures

Part 2	Network Monitoring Using SNMP	
Chapter 6	Configuring SNMPv3	101
	Figure 1: Inform Request and Response	132
Chapter 7	Configuring SNMP for Routing Instances	143
	Figure 2: SNMP Data for Routing Instances	144
Part 3	Remote Monitoring (RMON) with SNMP	
Chapter 14	Using RMON to Monitor Network Service Quality	235
	Figure 3: Setting Thresholds	236
	Figure 4: Network Entry Points	240
	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 the Documentation	xix
	Table 1: Notice Icons	xxi
	Table 2: Text and Syntax Conventions	xxii
Part 1	Overview	
Chapter 1	Network Management Overview	3
	Table 3: Device Management Features in Junos OS	4
Part 2	Network Monitoring Using SNMP	
Chapter 3	SNMP MIBs and Traps Supported by Junos OS	15
	Table 4: Standard MIBs Supported on Devices Running Junos OS	16
	Table 5: Enterprise-Specific MIBs and Supported Devices	46
Chapter 6	Configuring SNMPv3	101
	Table 6: Values to Use in Example	134
Chapter 7	Configuring SNMP for Routing Instances	143
	Table 7: MIB Support for Routing Instances (Juniper Networks MIBs)	144
	Table 8: Class 1 MIB Objects (Standard and Juniper MIBs)	148
	Table 9: Class 2 MIB Objects (Standard and Juniper MIBs)	152
	Table 10: Class 3 MIB Objects (Standard and Juniper MIBs)	153
	Table 11: Class 4 MIB Objects (Standard and Juniper MIBs)	154
Chapter 8	Configuring SNMP Remote Operations	161
	Table 12: Results in pingProbeHistoryTable: After the First Ping Test	169
	Table 13: Results in pingProbeHistoryTable: After the First Probe of the Second Test	170
	Table 14: Results in pingProbeHistoryTable: After the Second Ping Test	170
	Table 15: traceRouteProbeHistoryTable	178
Chapter 9	Tracing SNMP Activity	181
	Table 16: SNMP Tracing Flags	183
Chapter 10	SNMP FAQs	187
	Table 17: Monitored Object Instances	195
Part 3	Remote Monitoring (RMON) with SNMP	
Chapter 14	Using RMON to Monitor Network Service Quality	235
	Table 18: RMON Event Table	237
	Table 19: RMON Alarm Table	237

	Table 20: jnxRmon Alarm Extensions	238
	Table 21: Real-Time Performance Monitoring Configuration Options	244
	Table 22: Health Metrics	246
	Table 23: Counter Values for vlan-ccc Encapsulation	252
	Table 24: Performance Metrics	253
	Table 25: Inbound Traffic Per Class	257
	Table 26: Inbound Counters	257
	Table 27: Outbound Counters for ATM Interfaces	258
	Table 28: Outbound Counters for Non-ATM Interfaces	258
	Table 29: Dropped Traffic Counters	258
Part 4	Health Monitoring with SNMP	
Chapter 15	Configuring Health Monitoring	263
	Table 30: Monitored Object Instances	264
Part 5	Gathering Statistics for Accounting Purposes Using Accounting Options, Source Class Usage and Destination Class Usage Options	
Chapter 16	Accounting Options, Source Class Usage and Destination Class Usage Options Overview	269
	Table 31: Types of Accounting Profiles	269
Part 6	Configuration Statements and Operational Commands	
Chapter 23	Operational Commands	411
	Table 32: show ilmi statistics Output Fields	424
	Table 33: show snmp health-monitor Output Fields	426
	Table 34: show snmp inform-statistics Output Fields	433
	Table 35: show snmp mib Output Fields	436
	Table 36: show snmp rmon Output Fields	438
	Table 37: show snmp statistics Output Fields	442
	Table 38: show snmp v3 Output Fields	447

About the Documentation

- [Documentation and Release Notes on page xix](#)
- [Supported Platforms on page xix](#)
- [Using the Examples in This Manual on page xix](#)
- [Documentation Conventions on page xxi](#)
- [Documentation Feedback on page xxiii](#)
- [Requesting Technical Support on page xxiii](#)

Documentation and Release Notes

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

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

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

Supported Platforms

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

- [ACX Series](#)
- [M Series](#)
- [MX Series](#)
- [T Series](#)
- [PTX Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming

configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons







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

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

Table 2: Text and Syntax Conventions

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

GUI Conventions

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

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Network Management Overview on page 3](#)

CHAPTER 1

Network Management Overview

- [Understanding Device Management Functions in Junos OS on page 3](#)
- [Understanding the Integrated Local Management Interface on page 6](#)

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[®] operating system (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 CLI Explorer. SNMP MIBs—For more information about SNMP MIBs supported by Junos OS, see ““Standard SNMP MIBs Supported by Junos OS” on page 15” and ““Juniper Networks Enterprise-Specific MIBs” on page 34”. Standard SNMP traps—For more information about standard SNMP traps, see ““Standard SNMP Traps Supported on Devices Running Junos OS” on page 62”. Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see ““Juniper Networks Enterprise-Specific SNMP Traps” on page 63”. System log messages—For more information about how to configure system log messages, see the <i>Junos OS Administration Library for Routing Devices</i>. For more information about how to view system log messages, see the <i>Junos OS System Log Messages Reference</i>.
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 <i>Junos OS Administration Library for Routing Devices</i>. For more information about configuring the router using the APIs, see the <i>Junos XML Management Protocol Guide</i> and <i>NETCONF XML Management Protocol Guide</i>. Configuration Management MIB—For more information about the Configuration Management MIB, see the <i>Configuration Management MIB</i> in the <i>SNMP MIBs and Traps Reference</i>.

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 274. 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 <i>SNMP MIBs and Traps Reference</i>. 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 <i>Destination Class Usage MIB</i> and <i>Source Class Usage MIB</i> in the <i>SNMP MIBs and Traps Reference</i>, “Configuring Class Usage Profiles” on page 291, the <i>Junos OS Network Interfaces Library for Routing Devices</i>, and the <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i>. Count packets as part of a firewall filter. For more information about firewall filter policies, see “Juniper Networks Enterprise-Specific MIBs” on page 34 and the <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i>. 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 <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i>.
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 CLI Explorer. Use firewall filter. For more information about performance monitoring using firewall filters, see the <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i>. 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 <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i>. Use the enterprise-specific Class-of-Service MIB. For more information about this MIB, see the <i>Class-of-Service MIB</i> in the <i>SNMP MIBs and Traps Reference</i>.

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

Task	Junos OS Feature
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 <i>Junos OS Administration Library for Routing Devices</i>. Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see “Configuring the Local Engine ID” on page 108 and “Tracing SNMP Activity on a Device Running Junos OS” on page 181.

- Related Documentation**
- [Understanding the Integrated Local Management Interface on page 6](#)
 - [Understanding the SNMP Implementation in Junos OS on page 9](#)
 - [Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 239](#)
 - [Accounting Options Overview on page 269](#)

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 Library for Routing Devices*. For information about displaying ILMI statistics, see the [CLI Explorer](#). 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 2

Network Monitoring Using SNMP

- [SNMP Overview on page 9](#)
- [SNMP MIBs and Traps Supported by Junos OS on page 15](#)
- [Loading MIB Files to a Network Management System on page 65](#)
- [Configuring SNMP on page 69](#)
- [Configuring SNMPv3 on page 101](#)
- [Configuring SNMP for Routing Instances on page 143](#)
- [Configuring SNMP Remote Operations on page 161](#)
- [Tracing SNMP Activity on page 181](#)
- [SNMP FAQs on page 187](#)

CHAPTER 2

SNMP Overview

- [Understanding the SNMP Implementation in Junos OS on page 9](#)
- [SNMPv3 Overview on page 12](#)

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 Junos[®] operating system (Junos OS).

This topic includes the following sections:

- [SNMP Architecture on page 9](#)
- [Junos OS SNMP Agent Features on page 12](#)

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 10](#)
- [SNMP Traps and Informs on page 10](#)

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” on page 15](#).

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

For a list of Juniper Networks enterprise-specific supported MIBs, see [“Juniper Networks Enterprise-Specific MIBs” on page 34](#).

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 Junos OS, see [“Standard SNMP Traps Supported on Devices Running Junos OS” on page 62](#).

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 Junos OS, see [“Juniper Networks Enterprise-Specific SNMP Traps” on page 63](#). For information about system

logging severity levels for SNMP traps, see [“System Logging Severity Levels for SNMP Traps” on page 11](#).

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 131](#).

SNMP Trap Queuing

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.

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. 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.

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 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 Administration Library for Routing Devices* document.

For more information about system logging severity levels for standard traps, see *Standard SNMP Version 1 Traps* and *Standard SNMP Version 2 Traps*. 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*.

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.

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, 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 12](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 109](#)
- [Configuring MIB Views on page 97](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 121](#)
- [Configuring SNMP Informs on page 131](#)

**Related
Documentation**

- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

CHAPTER 3

SNMP MIBs and Traps Supported by Junos OS

- Standard SNMP MIBs Supported by Junos OS on page 15
- Juniper Networks Enterprise-Specific MIBs on page 34
- Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 46
- SNMP MIB Objects Supported by Junos OS for the Set Operation on page 55
- Standard SNMP Traps Supported on Devices Running Junos OS on page 62
- Juniper Networks Enterprise-Specific SNMP Traps on page 63

Standard SNMP MIBs Supported by Junos OS

Table 4 on page 16 contains the list of standard SNMP MIBs and RFCs that are supported on various devices running Junos OS. RFCs can be found at <http://www.ietf.org>.



NOTE: In this table, a value of 1 in any of the platform columns (ACX, M, T, J, MX, EX, PTX, and SRX) denotes that the corresponding MIB is supported on that particular platform, and a value of 0 denotes that the MIB is not supported on the platform.

Table 4: Standard MIBs Supported on Devices Running Junos OS

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
IEEE 802.1ab section 12.1, <i>Link Layer Discovery Protocol (LLDP) MIB</i>	0	0	0	0	1	1	0	0	0	0
EX Series implementation of LLDP MIB supports both IPv4 and IPv6 configuration.										
For more information about LLDP MIB objects supported on EX Series devices, see <i>LLDP Standard MIB Objects Supported on EX Series Devices</i> .										
IEEE, 802.3ad, <i>Aggregation of Multiple Link Segments</i>	0	1	1	1	1	1	1	1	1	1
Supported tables and objects:										
<ul style="list-style-type: none"> dot3adAggPortTable, dot3adAggPortListTable, dot3adAggTable, and dot3adAggPortStatsTable 										
NOTE: EX Series switches do not support the dot3adAggPortTable and dot3adAggPortStatsTable.										
<ul style="list-style-type: none"> dot3adAggPortDebugTable (only dot3adAggPortDebugRxState, dot3adAggPortDebugMuxState, dot3adAggPortDebugActorSyncTransitionCount, dot3adAggPortDebugPartnerSyncTransitionCount, dot3adAggPortDebugActorChangeCount, and dot3adAggPortDebugPartnerChangeCount) 										
NOTE: EX Series switches do not support the dot3adAggPortDebugTable.										
<ul style="list-style-type: none"> dot3adTablesLastChanged 										
NOTE: Gigabit Ethernet interfaces on J Series Services Routers do not support the 802.3ad MIB.										

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
IEEE, 802.1ag, <i>Connectivity Fault Management</i>	0	0	0	0	1	0	0	0	0	

Supported tables and objects:

- dot1agCfmMdTableNextIndex
- dot1agCfmMdTable (except dot1agCfmMdMhfdPermission)
- dot1agCfmMaNetTable
- dot1agCfmMaMepListTable
- dot1agCfmDefaultMdDefLevel
- dot1agCfmDefaultMdDefMhfCreation
- dot1agCfmMepTable (except dot1agCfmMepLbrBadMsdu, dot1agCfmMepTransmitLbmVlanPriority, dot1agCfmMepTransmitLbmVlanDropEnable, dot1agCfmMepTransmitLtmFlags, dot1agCfmMepPbbTeCanReportPbbTePresence, dot1agCfmMepPbbTeTrafficMismatchDefect, dot1agCfmMepPbbTransmitLbmLtmReverseVid, dot1agCfmMepPbbTeMismatchAlarm, dot1agCfmMepPbbTeLocalMismatchDefect, and dot1agCfmMepPbbTeMismatchSinceReset)
- dot1agCfmLtrTable (except dot1agCfmLtrChassisIdSubtype, dot1agCfmLtrChassisId, dot1agCfmLtrManAddressDomain, dot1agCfmLtrManAddress, dot1agCfmLtrIngressPortIdSubtype, dot1agCfmLtrIngressPortId, dot1agCfmLtrEgressPortIdSubtype, dot1agCfmLtrEgressPortId, and dot1agCfmLtrOrganizationSpecificTlv)
- dot1agCfmMepDbTable (except dot1agCfmMebDbChassisIdSubtype, dot1agCfmMebDbChassisId, dot1agCfmMebDbManAddressDomain, and dot1agCfmMebDbManAddress)

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
IEEE, 802.1ap, <i>Management Information Base (MIB) definitions for VLAN Bridges</i>	0	0	0	0	1	0	0	0	0	
Supported tables and objects:										
<ul style="list-style-type: none"> • ieee8021CfmStackTable • ieee8021CfmVlanTable • ieee8021CfmDefaultMdTable (except ieee8021CfmDefaultMdIdPermission) • ieee8021CfmMaCompTable (except ieee8021CfmMaCompIdPermission) 										
RFC 1155, <i>Structure and Identification of Management Information for TCP/IP-based Internets</i>	1	1	1	1	1	1	1	1	1	1
RFC 1157, <i>A Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	1	1	1
RFC 1195, <i>Use of OSI IS-IS for Routing in TCP/IP and Dual Environments</i> (only the objects isisSystem, isisMANAreaAddr, isisAreaAddr, isisSysProtSupp, isisSummAddr, isisCirc, isisCircLevel, isisPacketCount, isisISAdj, isisISAdjAreaAddr, isisAdjIPAddr, isisISAdjProtSupp, isisRa, and isisIPRA are supported)	1	1	1	1	1	1	1	1	1	1
RFC 1212, <i>Concise MIB Definitions</i>	1	1	1	1	1	1	1	0	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 1213, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> . Junos OS supports the following areas:	1	1	1	1	1	1	1	0	0	1
<ul style="list-style-type: none"> MIB II and its SNMP version 2 derivatives, including: <ul style="list-style-type: none"> Statistics counters IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, <i>IP Forwarding Table MIB</i>) SNMP management Interface management SNMPv1 Get, GetNext requests, and version 2 GetBulk request Junos OS-specific secured access list Master configuration keywords Reconfigurations upon SIGHUP 										
RFC 1215, <i>A Convention for Defining Traps for use with the SNMP</i> (only MIB II SNMP version 1 traps and version 2 notifications)	1	1	1	1	1	1	1	0	0	1
RFC 1406, <i>Definitions of Managed Objects for the DS1 and E1 Interface Types</i> (T1 MIB is supported)	1	1	1	1	0	0	0	1	0	0
RFC 1407, <i>Definitions of Managed Objects for the DS3/E3 Interface Type</i> (T3 MIB is supported)	0	1	1	1	0	0	0	0	0	0
RFC 1471, <i>Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol</i> (only pppLink group is supported. The pppLink group consists of the pppLcp 1 object and the tables pppLinkStatustable and pppLinkConfigTable).	0	1	0	0	1	0	1	0	0	0
RFC 1657, <i>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2</i>	1	1	1	1	1	1	0	0	0	0
RFC 1695, <i>Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2</i>	1	1	1	1	0	0	1	0	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 1850, <i>OSPF Version 2 Management Information Base</i> (except for the ospfOriginateNewLsas and ospfRxNewLsas objects, the Host Table, and the traps ospfOriginateLSA , ospfLsdbOverflow , and ospfLsdbApproachingOverflow)	1	1	1	1	1	1	1	1	0	0
RFC 1901, <i>Introduction to Community-based SNMPv2</i>	1	1	1	1	1	1	1	1	1	1
RFC 2011, <i>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</i>	1	1	1	1	1	1	1	0	0	0
RFC 2012, <i>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</i>	1	1	1	1	1	1	1	1	0	1
RFC 2013, <i>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</i>	1	1	1	1	1	1	1	1	0	1
RFC 2024, <i>Definitions of Managed Objects for Data Link Switching Using SMIv2</i> (except for the dlswInterface and dlswSdlc object groups; the dlswDirLocateMacTable , dlswDirNBTable , and dlswDirLocateNBTable tables; the dlswCircuitDiscReasonLocal and dlswCircuitDiscReasonRemote tabular objects; and the dlswDirMacCacheNextIndex and dlswDirNBCacheNextIndex scalar objects; read-only access)	0	1	1	1	1	0	0	0	0	0
RFC 2096, <i>IP Forwarding Table MIB</i> (The ipCidrRouteTable has been extended to include the tunnel name when the next hop is through an RSVP-signaled LSP.) NOTE: RFC 2096 has been replaced by RFC 4292. However, Junos OS currently supports both RFC 2096 and RFC 4292.	1	1	1	1	1	1	1	0	0	1
RFC 2115, <i>Management Information Base for Frame Relay DTEs Using SMIv2</i> (frDlcmiTable only; frCircuitTable and frErrTable are not supported)	0	1	1	1	1	0	0	1	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 2233, <i>The Interfaces Group MIB Using SMIv2</i>	1	1	1	1	1	1	1	1	0	1
NOTE: RFC 2233 has been replaced by RFC 2863, IF MIB. However, Junos OS supports both RFC 2233 and RFC 2863.										
RFC 2287, <i>Definitions of System-Level Managed Objects for Applications</i> (only the objects sysApplInstallPkgTable , sysApplInstallElmtTable , sysApplElmtRunTable , and sysApplMapTable)	1	1	1	1	1	1	1	1	0	1
RFC 2465, <i>Management Information Base for IP Version 6: Textual Conventions and General Group</i> (except for IPv6 interface statistics)	1	1	1	1	1	0	1	1	0	0
RFC 2495, <i>Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types</i> (except for dsx1FarEndConfigTable , dsx1FarEndCurrentTable , dsx1FarEndIntervalTable , dsx1FarEndTotalTable , and dsx1FracTable)	1	1	1	1	0	0	0	1	0	0
RFC 2515, <i>Definitions of Managed Objects for ATM Management</i> (except atmVpCrossConnectTable , atmVcCrossConnectTable , and aal5VccTable)	1	1	1	1	0	0	0	0	0	0
RFC 2570, <i>Introduction to Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1	1	1	0	0	1
RFC 2571, <i>An Architecture for Describing SNMP Management Frameworks</i> (read-only access)	1	1	1	1	1	1	1	1	0	1
NOTE: RFC 2571 has been replaced by RFC 3411. However, Junos OS supports both RFC 2571 and RFC 3411.										
RFC 2572, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> (read-only access)	1	1	1	1	1	1	1	1	0	1
NOTE: RFC 2572 has been replaced by RFC 3412. However, Junos OS supports both RFC 2572 and RFC 3412.										

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 2576, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1	1	1	1	0	1
<p>NOTE: RFC 2576 has been replaced by RFC 3584. However, Junos OS supports both RFC 2576 and RFC 3584.</p>										
RFC 2578, <i>Structure of Management Information Version 2 (SMIv2)</i>	1	1	1	1	1	1	1	0	0	1
RFC 2579, <i>Textual Conventions for SMIv2</i>	1	1	1	1	1	1	1	0	0	1
RFC 2580, <i>Conformance Statements for SMIv2</i>	1	1	1	1	1	1	1	0	0	1
RFC 2662, <i>Definitions of Managed Objects for ADSL Lines</i> (J Series Services Routers. All MIB tables, objects, and traps are applicable for the ADSL ATU-R agent.)	0	1	1	1	1	0	0	1	0	0
RFC 2665, <i>Definitions of Managed Objects for the Ethernet-like Interface Types</i>	1	1	1	1	1	1	1	1	0	1
<p>NOTE: For M, T and MX Series, the SNMP counters do not count the Ethernet header and frame check sequence (FCS). Therefore, the Ethernet header bytes and the FCS bytes are not included in the following four OIDs:</p> <ul style="list-style-type: none"> • ifInOctets • ifOutOctets • ifHCInOctets • ifHCOctets <p>However, the EX switches adhere to RFC 2665.</p>										
RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> (except row creation, the Set operation, and the object vrpStatsPacketLengthErrors)	1	1	1	1	1	1	1	1	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 2790, <i>Host Resources MIB</i>	1	1	1	1	1	1	1	1	0	1
<ul style="list-style-type: none"> Only the hrStorageTable. The file systems /, /config, /var, and /tmp always return the same index number. When SNMP restarts, the index numbers for the remaining file systems might change. Only the objects of the hrSystem and hrSWInstalled groups. 										
RFC 2819, <i>Remote Network Monitoring Management Information Base</i>	1	1	1	1	1	1	1	1	0	1
<ul style="list-style-type: none"> etherStatsTable (for Ethernet interfaces only), alarmTable, eventTable, and logTable are supported on all devices running Junos OS. historyControlTable and etherHistoryTable (except etherHistoryUtilization object) are supported only on EX Series switches. 										
RFC 2863, <i>The Interfaces Group MIB</i>	1	1	1	1	1	1	1	0	0	1
NOTE: RFC 2863 replaces RFC 2233. However, Junos OS supports both RFC 2233 and RFC 2863.										
RFC 2864, <i>The Inverted Stack Table Extension to the Interfaces Group MIB</i>	0	1	1	1	1	0	1	0	0	1
RFC 2922, <i>The Physical Topology (PTOPO) MIB</i>	0	0	0	0	0	1	0	1	0	1
Supported objects:										
ptopoConnDiscAlgorithm, ptopoConnAgentNetAddrType, ptopoConnAgentNetAddr, ptopoConnMultiMacSASeen, ptopoConnMultiNetSASeen, ptopoConnsStatic, ptopoConnLastVerifyTime, ptopoConnRowStatus										
RFC 2925, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i> (only the objects pingCtlTable , pingResultsTable , pingProbeHistoryTable , pingMaxConcurrentRequests , traceRouteCtlTable , traceRouteResultsTable , traceRouteProbeHistoryTable , and traceRouteHopsTable)	1	1	1	1	1	1	1	1	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 2932, <i>IPv4 Multicast Routing MIB</i>	1	1	1	1	1	1	1	1	0	1
RFC 2934, <i>Protocol Independent Multicast MIB for IPv4</i>	1	1	1	1	1	1	1	1	0	0
<p>NOTE: In Junos OS, RFC 2934 is implemented based on a draft version, <i>pimmib.mib</i>, of the now standard RFC.</p> <p>Support for the pimNeighborLoss trap was added in Release 11.4.</p>										
RFC 2981, <i>Event MIB</i>	1	1	1	1	1	0	1	0	0	0
RFC 3014, <i>Notification Log MIB</i>	1	1	1	1	1	0	1	0	0	0
RFC 3019, <i>IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol</i>	0	1	1	1	1	0	1	0	0	1
RFC 3410 <i>Introduction and Applicability Statements for Internet-Standard Management Framework</i>	1	1	1	1	1	1	1	0	0	1
RFC 3411, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i>	1	1	1	1	1	1	1	0	0	1
<p>NOTE: RFC 3411 replaces RFC 2571. However, Junos OS supports both RFC 3411 and RFC 2571.</p>										
RFC 3412, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	0	0	1
<p>NOTE: RFC 3412 replaces RFC 2572. However, Junos OS supports both RFC 3412 and RFC 2572.</p>										
RFC 3413, <i>Simple Network Management Protocol (SNMP) Applications</i> (except for the Proxy MIB)	1	1	1	1	1	1	1	1	0	1
RFC 3414, <i>User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)</i>	1	1	1	1	1	1	1	0	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 3415, <i>View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	0	0	1
RFC 3416, <i>Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	0	0	1
NOTE: RFC 3416 replaces RFC 1905, which was supported in earlier versions of Junos OS.										
RFC 3417, <i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	1	0	1
RFC 3418, <i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	0	0	1
NOTE: RFC 3418 replaces RFC 1907, which was supported in earlier versions of Junos OS.										
RFC 3498, <i>Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures</i> (implemented under the Juniper Networks enterprise branch [jnxExperiment])	0	1	1	1	0	0	0	0	0	0
RFC 3584 <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1	1	1	0	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 3591 <i>Managed Objects for the Optical Interface Type</i>	0	1	1	1	0	0	0	0	0	0
optIfOTMnTable (except optIfOTMnOpticalReach , optIfOTMnInterfaceType , and optIfOTMnOrder), optIfOChConfigTable (except optIfOChDirectionality and optIfOChCurrentStatus), optIfOTUkConfigTable (except optIfOTUkTraceIdentifierAccepted , optIfOTUkTIMDetMode , optIfOTUkTIMActEnabled , optIfOTUkTraceIdentifierTransmitted , optIfOTUkDEGThr , optIfOTUkDEGM , optIfOTUkSinkAdaptActive , and optIfOTUkSourceAdaptActive), and optIfODUkConfigTable (except optIfODUkPositionSeqCurrentSize and optIfODUkTtpPresent)										
RFC 3592, <i>Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type</i>	0	1	1	1	1	0	0	0	0	0
RFC 3621, <i>Power Ethernet MIB</i>	0	0	0	0	0	1	0	0	0	0
RFC 3637, <i>Definitions of Managed Objects for the Ethernet WAN Interface Sublayer</i> (except etherWisDeviceTable , etherWisSectionCurrentTable , and etherWisFarEndPathCurrentTable)	0	1	1	1	1	0	1	0	0	0
RFC 3811, <i>Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management</i>	1	1	1	1	1	0	1	1	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 3812, <i>Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)</i> (read only access) <ul style="list-style-type: none"> MPLS tunnels as interfaces are not supported. The following objects in the TunnelResource table are not supported: mplsTunnelResourceMeanRate, mplsTunnelResourceMaxBurstSize, mplsTunnelResourceMeanBurstSize, mplsTunnelResourceExBurstSize, mplsTunnelResourceWeight. mplsTunnelPerfTable and mplsTunnelCRLDPResTable are not supported. mplsTunnelCHopTable is supported on ingress routers only. <p>NOTE: The branch used by the proprietary LDP MIB (ldpmib.mib) conflicts with RFC 3812. ldpmib.mib has been deprecated and replaced by jnx-mpls-ldp.mib.</p>	1	1	1	1	1	0	1	0	0	0
RFC 3813, <i>Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)</i> (read-only access). mplsInterfacePerfTable , mplsInSegmentPerfTable , mplsOutSegmentPerfTable , mplsInSegmentMapTable , mplsXCUp , and mplsXCDown are not supported.	1	1	1	1	1	0	1	1	0	0
RFC 3826, <i>The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model</i>	1	1	1	1	1	1	1	0	0	1
RFC 3877, <i>Alarm Management Information Base</i> barring the exceptions listed here: <ul style="list-style-type: none"> Junos OS does not support the alarmActiveStatsTable. Traps that do not conform to the alarm model are not supported. However, these traps can be redefined to conform to the alarm model. 	0	0	0	0	1	0	0	0	0	

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 3896, <i>Definitions of Managed Objects for the DS3/E3 Interface Type</i> (except dsx3FarEndConfigTable , dsx3FarEndCurrentTable , dsx3FarEndIntervalTable , dsx3FarEndTotalTable , and dsx3FracTable)	0	1	1	1	0	0	0	0	0	0
RFC 4087, <i>IP Tunnel MIB</i> —Describes MIB objects in the following tables for managing tunnels of any type over IPv4 and IPv6 networks: <ul style="list-style-type: none"> tunnelIfTable—Provides information about the tunnels known to a router. tunnelInetConfigTable—Assists dynamic creation of tunnels and provides mapping from end-point addresses to the current interface index value. <p>NOTE: Junos OS supports MAX-ACCESS of read-only for all the MIB objects in tunnelIfTable and tunnelInetConfigTable tables.</p>	0	1	1	0	1	0	0	0	0	0
RFC 4188, <i>Definitions of Managed Objects for Bridges</i> —Supports 802.1D STP(1998). Supports only the following subtrees and objects: <ul style="list-style-type: none"> dot1dStp subtree is supported on MX Series 3D Universal Edge Routers. dot1dTpFdbAddress, dot1dTpFdbPort, and dot1dTpFdbStatus objects from the dot1dTpFdbTable of the dot1dTp subtree are supported on EX Series Ethernet Switches. <p>NOTE: dot1dTpLearnedEntryDiscards and dot1dTpAgingTime objects are supported on M and T Series routers.</p>	0	0	0	0	1	1	0	0	0	0
RFC 4273, <i>Definitions of Managed Objects for BGP-4</i> (only jnxBgpM2PrefixInPrefixes , jnxBgpM2PrefixInPrefixesAccepted , and jnxBgpM2PrefixInPrefixesRejected objects)	1	1	1	1	1	1	0	0	0	1

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 4292, <i>IP Forwarding MIB</i> — Describes a table and MIB objects for forwarding IP packets that are version independent: <ul style="list-style-type: none"> • inetCidrRouteTable—Provides the ability to display IP version-independent multipath CIDR routes and obsoletes the ipCidrRouteTable object. • inetCidrRouteNumber—Indicates the number of current routes and obsoletes the ipCidrRouteNumber object. • inetCidrRouteDiscards—Counts the number of valid routes that are discarded from inetCidrRouteTable and obsoletes the ipCidrRouteDiscards object. <p>NOTE: Junos OS currently supports these MIB objects that will be deprecated in future releases: ipCidrRouteTable, ipCidrRouteNumber, and ipCidrRouteDiscards.</p>	1	1	1	1	1	1	1	0	0	0
RFC 4293, <i>Management Information Base for the Internet Protocol (IP)</i> — Supports only the mandatory groups. For detailed information, see Standard IPv4/IPv6 MIBs .	0	0	0	0	1	1	0	0	0	0
RFC 4318, <i>Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol</i> —Supports 802.1w and 802.1t extensions for RSTP.	0	1	1	1	1	1	0	0	0	0
RFC 4363b, <i>Q-Bridge VLAN MIB</i>	0	0	0	0	1	1	0	0	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 4382 <i>MPLS/BGP Layer 3 Virtual Private Network (VPN) MIB</i>	0	1	1	1	1	1	1	0	0	0
<p>Junos OS support for RFC 4382 includes the following scalar objects and tables:</p> <ul style="list-style-type: none"> • <code>mplsL3VpnActiveVrfs</code> • <code>mplsL3VpnConfiguredVrfs</code> • <code>mplsL3VpnConnectedInterfaces</code> • <code>mplsL3VpnVrfConfMidRteThresh</code> • <code>mplsL3VpnVrfConfHighRteThresh</code> • <code>mplsL3VpnIfConfRowStatus</code> • <code>mplsL3VpnIILblRcvThrsh</code> • <code>mplsL3VpnNotificationEnable</code> • <code>mplsL3VpnVrfConfMaxPossRts</code> • <code>mplsL3VpnVrfConfRteMxThrshTime</code> • <code>mplsL3VpnVrfOperStatus</code> • <code>mplsL3VpnVrfPerfCurrNumRoutes</code> • <code>mplsL3VpnVrfPerfTable</code> • <code>mplsL3VpnVrfRteTable</code> • <code>mplsVpnVrfRTTable</code> • <code>mplsL3VpnVrfTable</code> <p>NOTE: The <code>mplsL3VpnIfConfTable</code> has not been implemented in the MPLS/BGP Layer 3 Virtual Private Network (VPN) MIB, because of limited utility and difficulty in representing the <code>DistProtocol</code> bit accurately.</p>										
RFC 4444, <i>IS-IS MIB</i>	1	1	1	1	1	1	1	1	0	0
RFC 4668, <i>RADIUS Accounting Client Management Information Base (MIB) for IPv6</i> (read-only access)	0	0	0	0	1	0	0	0	0	0
RFC 4670, <i>RADIUS Accounting Client Management Information Base (MIB)</i> (read-only access)	0	0	0	0	1	0	0	0	0	0
RFC 4801, <i>Definitions of Textual Conventions for Generalized Multiprotocol Label Switching (GMPLS) Management Information Base (MIB)</i> (read-only access)	0	1	1	1	1	0	0	0	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 4802, <i>Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering (TE) Management Information Base (MIB)</i> (read-only access). gmplsTunnelReversePerfTable , gmplsTeScalars , gmplsTunnelTable , gmplsTunnelARHopTable , gmplsTunnelCHopTable , and gmplsTunnelErrorTable are not supported.)	0	1	1	1	1	0	0	0	0	0
RFC 4803, <i>Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) Management Information Base (MIB)</i> (read-only access). gmplsLabelTable and gmplsOutsegmentTable are not supported.	0	1	1	1	1	0	0	0	0	0
NOTE: The tables in GMPLS TE (RFC 4802) and LSR (RFC 4803) MIBs are extensions of the corresponding tables from the MPLS TE (RFC 3812) and LSR (RFC 3813) MIBs and use the same index as the MPLS MIB tables.										

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RFC 5643, <i>Management Information Base for OSPFv3</i>	0	1	1	1	1	0	1	0	0	1
<p>NOTE: Junos OS support for this MIB is read-only.</p> <p>Junos OS does not support the following tables and objects defined in this MIB.</p> <ul style="list-style-type: none"> ospfv3HostTable ospfv3CfgNbrTable ospfv3ExitOverflowInterval ospfv3ReferenceBandwidth ospfv3RestartSupport ospfv3RestartInterval ospfv3RestartStrictLsaChecking ospfv3RestartStatus ospfv3RestartAge ospfv3RestartExitReason ospfv3NotificationEnable ospfv3StubRouterSupport ospfv3StubRouterAdvertisement ospfv3DiscontinuityTime ospfv3RestartTime ospfv3AreaNssaTranslatorRole ospfv3AreaNssaTranslatorState ospfv3AreaNssaTranslatorStabInterval ospfv3AreaNssaTranslatorEvents ospfv3AreaTEEnabled ospfv3IfMetricValue ospfv3IfDemandNbrProbe 										
RFC 6527, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)</i> (except row creation, the Set operation, and the objects vrrpv3StatisticsRowDiscontinuityTime and vrrpv3StatisticsPacketLengthErrors)	1	0	0	1	0	0	0	0	0	0
Internet Assigned Numbers Authority, <i>IANAiftype Textual Convention MIB</i> (referenced by RFC 2233, available at http://www.iana.org/assignments/ianaiftype-mib)	1	1	1	1	1	1	1	1	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Internet draft draft-ietf-atommib-sonetaps-mib-10.txt, <i>Definitions of Managed Objects for SONET Linear APS Architectures</i> (as defined under the Juniper Networks enterprise branch [jnxExperiment] only)	0	1	1	1	1	0	0	0	0	0
Internet draft draft-ietf-bfd-mib-02.txt, <i>Bidirectional Forwarding Detection Management Information Base</i> (Represented by mib-jnx-bfd-exp.txt and implemented under the Juniper Networks enterprise branch [jnxExperiment]. Read only. Includes bfdSessUp and bfdSessDown traps. Does not support bfdSessPerfTable and bfdSessMapTable .)	1	1	1	1	1	1	0	0	0	1
Internet draft draft-ietf-idmr-igmp-mib-13.txt, <i>Internet Group Management Protocol (IGMP) MIB</i>	0	1	1	1	1	1	1	0	0	1
Internet draft draft-reeder-snmv3-usm-3desede-00.txt, <i>Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode</i>	1	1	1	1	1	1	1	0	0	1
Internet draft draft-ietf-isis-wg-mib-07.txt, <i>Management Information Base for IS-IS</i> (only isisSAdjTable , isisSAdjAreaAddrTable , isisSAdjIPAddrTable , and isisSAdjProtSuppTable) NOTE: Replaced with RFC 4444, <i>IS-IS MIB</i> in Junos OS Release 11.3 and later.	1	1	1	1	1	1	1	1	0	0
Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt, <i>MPLS/BGP Virtual Private Network Management Information Base Using SMIv2</i> (only mplsVpnScalars , mplsVpnVrfTable , mplsVpnPerTable , and mplsVpnVrfRouteTargetTable)	0	1	1	1	1	0	1	0	0	0

Table 4: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Internet draft draft-ietf-ospf-ospfv3-mib-11.txt, <i>Management Information Base for OSPFv3</i> (Represented by <code>mib-jnx-ospfv3mib.txt</code> and implemented under the Juniper Networks enterprise branch <code>{jnxExperiment}</code> . Support for <code>ospfv3NbrTable</code> only. Read only. Object names are prefixed by <code>jnx</code> . For example, <code>jnxOspfv3NbrTable</code> , <code>jnxOspfv3NbrAddressType</code> , and <code>jnxOspfv3NbrPriority</code> .)	0	1	1	1	1	0	1	0	0	1
Internet draft draft-ietf-idmr-pim-mib-09.txt, <i>Protocol Independent Multicast (PIM) MIB</i>	1	1	1	1	1	1	1	0	0	1
ESO Consortium MIB, which can be found at http://www.snmp.com/eso/ <i>NOTE:</i> The ESO Consortium MIB has been replaced by RFC 3826.	1	1	1	1	1	1	1	1	0	0
Internet Draft P2MP MPLS-TE MIB (draft-ietf-mpls-p2mp-te-mib-09.txt) (read-only access) (except <code>mplsTeP2mpTunnelBranchPerfTable</code>).	1	1	1	1	1	0	1	0	0	0

- Related Documentation**
- [Juniper Networks Enterprise-Specific MIBs on page 34](#)
 - [Loading MIB Files to a Network Management System on page 65](#)

Juniper Networks Enterprise-Specific MIBs

Junos OS supports the following enterprise-specific MIBs:

- **AAA Objects MIB**—Provides support for monitoring user authentication, authorization, and accounting through the RADIUS, LDAP, SecurID, and local authentication servers. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-user--aaa.txt. For more information, see *AAA Objects MIB*.
- **Access Authentication Objects MIB**—Provides support for monitoring firewall authentication, including data about the users trying to access firewall-protected resources and the firewall authentication service itself. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of

this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-auth.txt.

For more information, see *Access Authentication Objects MIB*.

- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis-alarm.txt

For more information, see *Alarm MIB*.

- Analyzer MIB—Contains analyzer and remote analyzer data related to port mirroring on the EX Series Ethernet Switches. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-analyzer.txt

For more information, see *Analyzer MIB*.

- Antivirus Objects MIB—Provides information about the antivirus engine, antivirus scans, and antivirus scan-related traps. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-utm-av.txt

For more information, see *Antivirus Objects MIB*.

- ATM Class-of-Service MIB—Provides support for monitoring Asynchronous Transfer Mode, version 2 (ATM2) virtual circuit (VC) class-of-service (CoS) configurations. It also provides CoS queue statistics for all VCs that have CoS configured. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-atm-cos.txt

For more information, see *ATM Class-of-Service MIB*.

- ATM MIB—Provides support for ATM interfaces and virtual connections. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-atm.txt.

For more information, see *ATM MIB*.

- BGP4 V2 MIB—Contains objects used to monitor BGP peer-received prefix counters. It is based upon similar objects in the MIB documented in Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of BGP (BGP-4), Second Version*. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-bgpmib2.txt

For more information, see *BGP4 V2 MIB*.

- Bidirectional Forwarding Detection MIB—Provides support for monitoring Bidirectional Forwarding Detection (BFD) sessions. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-bfd.txt

For more information, see *Bidirectional Forwarding Detection MIB*.

- Chassis Definitions for Router Model MIB—Contains the object identifiers (OIDs) that are used by the Chassis MIB to identify platform and chassis components. The Chassis MIB provides information that changes often, whereas the Chassis Definitions for Router Model MIB provides information that changes less often. For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chas-defines.txt

For more information, see *Chassis MIBs*.

- Chassis Forwarding MIB—Enables J Series Services Routers to fully support the Junos OS health monitor. This MIB extends the scope of health monitoring to include Junos forwarding process (**fwdd**) components. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis-fwdd.txt

For more information, see *Chassis Forwarding MIB*.

- Chassis MIB—Provides support for environmental monitoring (power supply state, board voltages, fans, temperatures, and air flow) and inventory support for the chassis, System Control Board (SCB), System and Switch Board (SSB), Switching and Forwarding Module (SFM), Flexible PIC Concentrators (FPCs), and PICs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis.txt

For more information, see *Chassis MIBs*.

- Chassis Cluster MIB—Provides information about objects that are used whenever the state of the control link interfaces or fabric link interfaces changes (up to down or down to up) in a chassis cluster deployment. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jsrpd.txt.

For more information, see *Chassis Cluster MIB*.

- Class-of-Service MIB—Provides support for monitoring interface output queue statistics per interface and per forwarding class. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-cos.txt

For more information, see *Class-of-Service MIB*.

- Configuration Management MIB—Provides notification for configuration changes as SNMP traps. Each trap contains the time at which the configuration change was committed, the name of the user who made the change, and the method by which the change was made. A history of the last 32 configuration changes is kept in **jnxCmChgEventTable**. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-cfgmgmt.txt

For more information, see *Configuration Management MIB*.

- Destination Class Usage MIB—Provides support for monitoring packet counts based on the ingress and egress points for traffic transiting your networks. Ingress points are identified by the input interface. Egress points are identified by destination prefixes grouped into one or more sets, known as destination classes. One counter is managed per interface per destination class, up to a maximum of 16 counters per interface. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dcu.txt

For more information, see *Destination Class Usage MIB*.

- DHCP Objects MIB— Provides SNMP support (get and trap) for DHCP local server and relay configurations. It also provides support for bindings and leases tables, and for statistics. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jdhcp.txt

For more information, see *DHCP MIB*.

- DHCPv6 MIB—Provides SNMP support (get and trap) for DHCPv6 local server and relay configurations. It also provides support for bindings and leases tables, and for statistics. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jdhcpv6.txt
For more information, see *DHCPv6 MIB*.
- Digital Optical Monitoring MIB—Provides support for the **SNMP Get** request for statistics and **SNMP Trap** notifications for alarms. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dom.txt
For more information, see *Digital Optical Monitoring MIB*.
- DNS Objects MIB—Provides support for monitoring DNS proxy queries, requests, responses, and failures. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-dns.txt
For more information, see *DNS Objects MIB*.
- Dynamic Flow Capture MIB—Provides support for monitoring the operational status of dynamic flow capture (DFC) PICs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dfc.txt
For more information, see *Dynamic Flow Capture MIB*.
- Ethernet MAC MIB—Monitors media access control (MAC) statistics on Gigabit Ethernet intelligent queuing (IQ) interfaces. It collects MAC statistics; for example, **inoctets**, **inframes**, **outoctets**, and **outframes** on each source MAC address and virtual LAN (VLAN) ID for each Ethernet port. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mac.txt
For more information, see *Ethernet MAC MIB*.
- Event MIB—Defines a generic trap that can be generated using an op script or event policy. This MIB provides the ability to specify a system log string and raise a trap if that system log string is found. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-event.txt
For more information, see *Event MIB*.
- Experimental MIB—Contains object identifiers for experimental MIBs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-exp.txt
For more information, see *jnxExperiment MIB*.
- Firewall MIB—Provides support for monitoring firewall filter counters. Routers must have the Internet Processor II ASIC to perform firewall monitoring. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-firewall.txt
For more information, see *Firewall MIB*.
- Flow Collection Services MIB—Provides statistics on files, records, memory, FTP, and error states of a monitoring services interface. It also provides SNMP traps for unavailable destinations, unsuccessful file transfers, flow overloading, and memory

overloading. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-coll.txt

For more information, see *Flow Collection Services MIB*.

- Host Resources MIB—Extends the **hrStorageTable** object, providing a measure of the usage of each file system on the router in percentage format. Previously, the objects in the **hrStorageTable** measured the usage in allocation units—**hrStorageUsed** and **hrStorageAllocationUnits**—only. Using the percentage measurement, you can more easily monitor and apply thresholds on usage. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-hostresources.txt

For more information, see *Host Resources MIB*.

- IDP Objects MIB—Provides support for monitoring SNMP IDP queries, requests, responses, and failures. This MIB defines the key monitoring and threshold crossing trap support, IDP database update status and trap support, attack-related monitoring and trap support for SRX100, SRX210, SRX220, SRX240, SRX550, and SRX650 Services Gateways. This MIB models IDP attributes specific to the appropriate Juniper Networks implementation. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-idp.txt

For more information, see *IDP MIB*.

- Interface MIB—Extends the standard **ifTable** (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-if-extensions.txt

For more information, see *Interface MIB*.

- Interface Accounting Forwarding Class MIB—Extends the Juniper Enterprise Interface MIB and provides support for monitoring statistics data for interface accounting and IETF standardization. This MIB is currently supported by Junos OS for M Series and MX Series devices only. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-if-accounting.txt

For more information, see *Interface Accounting Forwarding Class MIB*.

- IP Forward MIB—Extends the standard IP Forwarding Table MIB (RFC 4292) to include CIDR forwarding information. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipforward.txt

For more information, see *IP Forwarding MIB*.

- IPsec Generic Flow Monitoring Object MIB—Based on **jnx-ipsec-monitor-mib**, this MIB provides support for monitoring IPsec and IPsec VPN management objects. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt

For more information, see *IPsec Generic Flow Monitoring Object MIB*.

- IPsec Monitoring MIB—Provides operational and statistical information related to the IPsec and IKE tunnels on Juniper Networks routers. For a downloadable version of this

MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt

.

For more information, see *IPSec Monitoring MIB*.

- IPsec VPN Objects MIB—Provides support for monitoring IPsec and IPsec VPN management objects for Juniper security product lines. This MIB is an extension of **jnx-ipsec-flow-mon.mib**. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-ipsec-vpn.txt

For more information, see *IPsec VPN Objects MIB*.

- IPv4 MIB—Provides additional Internet Protocol version 4 (IPv4) address information, supporting the assignment of identical IPv4 addresses to separate interfaces. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipv4.txt

For more information, see *IPv4 MIB*.

- IPv6 and ICMPv6 MIB—Provides IPv6 and Internet Control Message Protocol version 6 (ICMPv6) statistics. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipv6.txt

For more information, see *IPv6 MIB*.

- L2ALD MIB—Contains information about the Layer 2 Address Learning Daemon (L2ALD) and related traps, such as the routing instance MAC limit trap and the interface MAC limit trap. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2ald.txt

For more information, see *L2ALD MIB*.

- L2CP MIB—Provides information about Layer 2 Control Protocols (L2CP) based features on MX Series 3D Universal Edge Routers. Currently, Junos OS supports only the **jnxDot1dStpPortRootProtectEnabled**, **jnxDot1dStpPortRootProtectState**, and **jnxPortRootProtectStateChangeTrap** objects. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2cp-features.txt

For more information, see *L2CP MIB*.

- L2TP MIB—Provides information about Layer 2 Transport Protocol (L2TP) tunnels and sessions. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2tp.txt

For more information, see *L2TP MIB*.

- LDP MIB—Provides LDP statistics and defines LDP label-switched path (LSP) notifications. LDP traps support only IPv4 standards. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ldp.txt

For more information, see *LDP MIB*.

- License MIB—Extends SNMP support to licensing information, and introduces SNMP traps that alert users when the licenses are about to expire, expire, or when the total number of users exceeds the number specified in the license. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-license.txt

For more information, see *License MIB*.

- Logical Systems MIBs—Extend SNMP support to logical systems security profile through various MIBs defined under **jnxLsysSecurityProfile**. For downloadable versions of the MIBs, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt

For more information, see *Logical Systems MIB*.

- MIMSTP MIB—Provides information about MSTP instances (that is, routing instances of type Virtual Switch/Layer 2 control, also known as virtual contexts), MSTIs within the MSTP instance, and VLANs associated with the MSTI. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mimstp.txt

For more information, see *MIMSTP MIB*.

- MPLS MIB—Provides MPLS information and defines MPLS notifications. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mpls.txt



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

For more information, see *MPLS MIB*.

- MPLS LDP MIB—Contains object definitions as described in RFC 3815, *Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)*. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mpls-ldp.txt



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

For more information, see *MPLS LDP MIB*.

- NAT Objects MIB—Provides support for monitoring network address translation (NAT). This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-nat.txt

For more information, see *NAT Objects MIB*.

- NAT Resources-Monitoring MIB—Provides support for monitoring NAT pools usage and NAT rules. Notifications of usage of NAT resources are also provided by this MIB. This MIB is currently supported on the Multiservices PIC and Multiservices DPC on M Series and MX Series routers only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sp-nat.txt

For more information, see *Network Address Translation Resources—Monitoring MIB*.

- OTN Interface Management MIB—Defines objects for managing Optical Transport Network (OTN) interfaces on devices running Junos OS. For a downloadable version of the MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-otn.txt

For more information, see *OTN Interface Management MIB*.

- Packet Forwarding Engine MIB—Provides notification statistics for Packet Forwarding Engines. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pfe.txt

For more information, see *Packet Forwarding Engine MIB*.

- Packet Mirror MIB—Enables you to capture and view packet mirroring-related information. This MIB is currently supported by Junos OS for MX Series routers only. Packet mirroring traps are an extension of the standard SNMP implementation and are only available to SNMPv3 users. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-packet-mirror.txt

For more information, see *Packet Mirror MIB Overview*.

- PAE Extension MIB—Extends the standard IEEE802.1x PAE Extension MIB, and contains information for Static MAC Authentication. The enterprise-specific PAE Extension MIB is supported only on EX Series Ethernet Switches. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pae-extension.txt

For more information, see *PAE Extension MIB*.

- Passive Monitoring MIB—Performs traffic flow monitoring and lawful interception of packets transiting between two routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pmon.txt

For more information, see *Passive Monitoring MIB*.

- Ping MIB—Extends the standard Ping MIB control table (RFC 2925). Items in this MIB are created when entries are created in **pingCtlTable** of the Ping MIB. Each item is indexed exactly as it is in the Ping MIB. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ping.txt

For more information, see *PING MIB*.

- Policy Objects MIB—Provides support for monitoring the security policies that control the flow of traffic from one zone to another. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-policy.txt

For more information, see *Policy Objects MIB*.

- Power Supply Unit MIB—Enables monitoring and managing of the power supply on a device running Junos OS. This MIB is currently supported only on EX Series Ethernet Switches. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-power-supply-unit.txt

For more information, see *Power Supply Unit MIB*.

- PPP MIB—Provides SNMP support for PPP-related information such as the type of authentication used, interface characteristics, status, and statistics. This MIB is supported on Common Edge PPP process, jpppd. This MIB is currently supported only on M Series and MX Series routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ppp.txt

For more information, see *PPP MIB*.

- PPPoE MIB—Provides SNMP support for PPPoE-related information such as the type of authentication used, interface characteristics, status, and statistics. This MIB is supported on Common Edge PPPoE process, jpppoed. This MIB is currently supported only on M Series and MX Series routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pppoe.txt

For more information, see *PPPoE MIB*.

- Pseudowire TDM MIB—Extends the standard Pseudowire MIB, and contains information about configuration and statistics for specific pseudowire types. The enterprise-specific Pseudowire TDM MIB is the Juniper Networks implementation of the standard Managed Objects for TDM over Packet Switched Network MIB (draft-ietf-pwe3-tdm-mib-08.txt). For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pwtdm.txt

For more information, see *Pseudowire TDM MIB*.

- Pseudowire ATM MIB—Extends the standard Pseudowire MIB, and defines objects used for managing the ATM pseudowires in Juniper products. The enterprise-specific Pseudowire ATM MIB is the Juniper Networks implementation of RFC 5605, *Managed Objects for ATM over Packet Switched Networks (PSNs)*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pwatm.txt

For more information, see *Pseudowire ATM MIB*.

- Real-Time Performance Monitoring MIB—Provides real-time performance-related data and enables you to access jitter measurements and calculations using SNMP. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rpm.txt

For more information, see *Real-Time Performance Monitoring MIB*.

- Reverse-Path-Forwarding MIB—Monitors statistics for traffic that is rejected because of reverse-path-forwarding (RPF) processing. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rpf.txt



NOTE: The enterprise-specific RPF MIB is not supported on EX Series Ethernet Switches.

For more information, see *Reverse Path Forwarding MIB*.

- RMON Events and Alarms MIB—Supports the Junos OS extensions to the standard Remote Monitoring (RMON) Events and Alarms MIB (RFC 2819). The extension augments **alarmTable** with additional information about each alarm. Two new traps are also defined to indicate when problems are encountered with an alarm. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rmon.txt

For more information, see *RMON Events and Alarms MIB*.

- RSVP MIB—Provides information about RSVP-traffic engineering sessions that correspond to MPLS LSPs on transit routers in the service provider core network. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rsvp.txt



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

For more information, see *RSVP MIB*.

- Security Interface Extension Objects MIB—Provides support for the security management of interfaces. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-if-ext.txt

For more information, see *Security Interface Extension Objects MIB*.

- Security Screening Objects MIB—Defines the MIB for the Juniper Networks Enterprise Firewall screen functionality. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-screening.txt

For more information, see *Security Screening Objects MIB*.

- Services PIC MIB—Provides statistics for Adaptive Services (AS) PICs and defines notifications for AS PICs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sp.txt

For more information, see *Services PIC MIB*.

- SONET APS MIB—Monitors any SONET interface that participates in Automatic Protection Switching (APS). For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sonetaps.txt

For more information, see *SONET APS MIB*.

- SONET/SDH Interface Management MIB—Monitors the current alarm for each SONET/SDH interface. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sonet.txt
For more information, see *SONET/SDH Interface Management MIB*.
- Source Class Usage MIB—Counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. The Source Class Usage (SCU) MIB makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-scu.txt
For more information, see *Source Class Usage MIB*.
- SPU Monitoring MIB—Provides support for monitoring SPUs on SRX5600 and SRX5800 devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt
For more information, see *SPU Monitoring Objects MIB*.
- Structure of Management Information MIB—Explains how the Juniper Networks enterprise-specific MIBs are structured. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-smi.txt
For more information, see *Structure of Management Information MIB*.
- Structure of Management Information MIB for EX Series Ethernet Switches—Defines a MIB branch for switching-related MIB definitions for the EX Series Ethernet Switches. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ex-smi.txt
For more information, see *EX Series SMI MIB*.
- Structure of Management Information MIB for J Series and SRX Series —Contains object identifiers (OIDs) for the security branch of the MIBs used in Junos OS for J Series and SRX Series devices, services, and traps. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-smi.txt
For more information, see *Structure of Management Information MIB*.
- Subscriber MIB—Provides SNMP support for subscriber-related information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-subscriber.txt
For more information, see *Subscriber MIB*.
- System Log MIB—Enables notification of an SNMP trap-based application when an important system log message occurs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-syslog.txt
For more information, see *System Log MIB*.
- Traceroute MIB—Supports the Junos OS extensions of traceroute and remote operations. Items in this MIB are created when entries are created in the

traceRouteCtlTable of the Traceroute MIB. Each item is indexed exactly the same way as it is in the Traceroute MIB. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-traceroute.txt

For more information, see *Traceroute MIB*.

- Utility MIB—Provides SNMP support for exposing the Junos OS data and has tables that contain information about each type of data, such as integer and string. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-util.txt

For more information, see *Utility MIB*.

- Virtual Chassis MIB—Contains information about the virtual chassis on the EX Series Ethernet Switches and the MX Series. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-virtualchassis.txt

For more information, see *Virtual Chassis MIBs*.

- VLAN MIB—Contains information about prestandard IEEE 802.10 VLANs and their association with LAN emulation clients. The enterprise-specific VLAN MIB is supported only on EX Series Ethernet Switches. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vlan.txt

For more information, see *VLAN MIB*.

- VPLS MIBs—Provides information about generic, BGP-based, and LDP-based VPLS and pseudowires associated with the VPLS networks. The enterprise-specific VPLS MIBs are Juniper Networks extensions of the following IETF standard MIBs defined in Internet draft draft-ietf-l2vpn-vpls-mib-05.txt, and are implemented as part of the **jnxExperiment** branch:

- **VPLS-Generic-Draft-01-MIB** implemented as **mib-jnx-vpls-generic.txt**

- **VPLS-BGP-Draft-01-MIB** implemented as **mib-jnx-vpls-bgp.txt**

- **VPLS-LDP-Draft-01-MIB** implemented as **mib-jnx-vpls-ldp.txt**

For downloadable versions of these MIBs, see:

- http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-generic.txt

- http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-bgp.txt

- http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-ldp.txt

For more information, see *Interpreting the Enterprise-Specific VPLS MIBs*.

- VPN Certificate Objects MIB—Provides support for monitoring the local and CA certificates loaded on the router. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-cert.txt

For more information, see *VPN Certificate Objects MIB*.

- VPN MIB—Provides monitoring for Layer 3 VPNs, Layer 2 VPNs, and virtual private LAN service (VPLS) (read access only). For a downloadable version of the MIB, see

http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpn.txt

For more information, see *VPN MIB*.

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 15](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 46](#)
 - [Loading MIB Files to a Network Management System on page 65](#)

Juniper Networks Enterprise-Specific MIBs and Supported Devices

Table 5 on page 46 lists the enterprise-specific MIBs that are supported on various devices running Junos OS.



NOTE: In this table, a value of 1 in any of the platform columns (ACX, M, MX, T, EX, J, and SRX) denotes that the corresponding MIB is supported on that particular platform. A value of 0 denotes that the MIB is not supported on the platform.



NOTE: This topic uses the following classification for SRX devices: Low-End (SRX100, SRX210, SRX220, and SRX240), Mid-Range (SRX550, SRX650 and SRX1400), and High-End (SRX3400, SRX3600, SRX5600, and SRX5800).

Table 5: Enterprise-Specific MIBs and Supported Devices

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
AAA Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-user-aaa.txt	0	1	1	0	0	0	0	0	1	1
Access Authentication Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-auth.txt	0	0	0	0	0	1	0	1	1	1
Alarm MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis-alarm.txt	1	1	1	1	1	1	1	1	1	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Analyzer MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-analyzer.txt	0	0	0	0	1	0	0	0	0	0
Antivirus Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-utm-av.txt	0	0	0	0	0	0	0	1	0	0
ATM Class-of-Service MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-atm-cos.txt	0	1	1	1	0	0	0	1	0	1
ATM MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-atm.txt	1	1	1	1	0	0	0	0	0	0
BGP4 V2 MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-bgpmib2.txt	1	1	1	1	1	1	1	1	1	1
Bidirectional Forwarding Detection MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-bfd.txt	1	1	1	1	1	1	1	1	1	1
Chassis Forwarding MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis-fwdd.txt	1	0	0	0	0	1	1	1	0	0
Chassis MIBs http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chassis.txt http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-chas-defines.txt	1	1	1	1	1	1	1	1	1	1
Chassis Cluster MIBs http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jsrpd.txt	0	0	0	0	0	0	0	0	1	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Class-of-Service MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-cos.txt	1	1	1	1	1	1	1	0	0	1
Configuration Management MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-cfgmgmt.txt	1	1	1	1	1	1	1	1	1	1
Destination Class Usage MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dcu.txt	0	1	1	1	0	1	0	0	1	1
DHCP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jdhcp.txt	0	1	1	1	0	0	0	0	0	0
DHCPv6 MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-jdhcpv6.txt	0	1	1	1	0	0	0	0	0	0
Digital Optical Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dom.txt	1	1	1	1	1	1	0	1	1	1
DNS Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-dns.txt	0	0	0	0	0	0	0	0	1	1
Dynamic Flow Capture MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-dfc.txt	0	1	1	1	0	0	0	0	0	0
Ethernet MAC MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/jnx-mac.txt	0	1	1	1	1	1	0	0	0	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Event MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-event.txt	1	1	1	1	1	1	1	1	1	1
EX Series MAC Notification MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ex-mac-notification.txt	0	0	0	0	1	0	0	0	0	0
EX Series SMI MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ex-smi.txt	0	0	0	0	1	0	0	0	0	0
Experimental MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-exp.txt	1	1	1	1	1	1	0	0	0	0
Firewall MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-firewall.txt	1	1	1	1	1	1	1	1	1	1
Flow Collection Services MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-coll.txt	0	1	1	1	0	0	0	0	0	0
Host Resources MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-hostresources.txt	1	1	1	1	1	1	0	1	1	1
Interface MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-if-extensions.txt	1	1	1	1	1	1	1	1	1	1
IP Forward MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipforward.txt	1	1	1	1	1	1	1	1	1	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
IPsec Generic Flow Monitoring Object MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt	0	0	0	1	0	0	0	1	1	1
IPsec Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt	0	1	1	1	0	1	0	0	0	0
IPsec VPN Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-ipsec-vpn.txt	0	0	0	1	0	0	0	1	1	1
IPv4 MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipv4.txt	1	1	1	1	1	1	1	1	1	1
IPv6 and ICMPv6 MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ipv6.txt	0	1	1	1	1	0	1	1	1	1
L2ALD MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2ald.txt	0	0	1	0	1	0	0	0	0	0
L2CP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2cp-features.txt	0	0	0	0	1	0	0	0	0	0
L2TP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-l2tp.txt	0	1	1	0	0	0	0	0	0	0
LDP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ldp.txt	1	1	1	1	0	0	1	0	0	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
License MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-license.txt	0	1	1	1	0	0	0	1	1	1
Logical Systems MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt	0	0	0	0	0	0	0	0	1	1
MIMSTP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mimstp.txt	0	0	1	0	1	0	0	0	0	0
MPLS LDP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mpls-ldp.txt	1	1	1	1	1	1	1	0	0	0
MPLS MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-mpls.txt	1	1	1	1	1	1	1	0	0	1
NAT Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-nat.txt	0	0	0	0	0	1	0	1	1	1
NAT Resources-Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sp-nat.txt	0	1	1	1	0	0	0	0	0	0
OTN Interface Management MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-otn.txt	0	1	1	1	0	0	0	0	0	0
Packet Forwarding Engine MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pfe.txt	1	1	1	1	0	1	1	1	1	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Packet Mirror MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-packet-mirror.txt	0	0	0	0	1	0	0	0	0	0
PAE Extension MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pae-extension.txt	0	0	0	0	1	0	0	0	0	0
Passive Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pmon.txt	0	1	1	1	0	0	0	0	0	0
Ping MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ping.txt	1	1	1	1	1	1	0	1	1	1
Policy Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-policy.txt	0	0	0	0	0	1	0	1	1	1
Power Supply Unit MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-power-supply-unit.txt	0	0	0	0	1	0	1	0	0	0
PPP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-ppp.txt	0	1	1	0	0	0	0	0	0	0
PPPoE MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pppoe.txt	0	1	1	0	0	0	0	0	0	0
Pseudowire ATM MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pwatm.txt	0	1	0	0	1	0	0	0	0	0

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Pseudowire TDM MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-pwtdm.txt	1	1	1	1	0	0	0	0	0	0
Real-Time Performance Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rpm.txt	0	1	1	1	1	1	0	1	0	0
Reverse-Path-Forwarding MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rpf.txt	1	1	1	1	1	1	1	1	1	1
RMON Events and Alarms MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rmon.txt	1	1	1	1	1	1	1	1	1	1
RSVP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-rsvp.txt	1	1	1	1	1	0	1	0	0	0
Security Interface Extension Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-if-ext.txt	0	0	0	0	0	1	0	1	1	1
Security Screening Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-screening.txt	0	0	0	0	0	0	0	0	0	1
Services PIC MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sp.txt	0	1	1	1	0	0	0	0	0	0
SNMP IDP MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-idp.txt	0	0	0	0	0	0	0	1	1	0

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
SONET APS MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sonetaps.txt	0	1	1	1	0	0	0	0	0	0
SONET/SDH Interface Management MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-sonet.txt	0	1	1	1	0	0	0	0	0	0
Source Class Usage MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-scu.txt	0	1	1	1	0	0	0	0	0	1
SPU Monitoring MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt	0	0	0	0	0	0	0	1	1	1
Structure of Management Information MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-smi.txt	1	1	1	1	1	1	0	1	1	1
Subscriber MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-subscriber.txt	1	0	1	0	0	0	0	0	0	0
System Log MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-syslog.txt	0	1	1	1	1	1	1	1	1	1
Traceroute MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-traceroute.txt	0	1	1	1	1	1	0	1	1	1
Utility MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-util.txt	0	1	1	1	1	1	0	1	1	1

Table 5: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Virtual Chassis MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-virtualchassis.txt	0	0	0	0	1	1	0	0	0	0
VLAN MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vlan.txt	0	0	0	0	1	0	0	0	0	0
VPLS MIBs <ul style="list-style-type: none"> http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-generic.txt http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-ldp.txt http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpls-bgp.txt 	0	1	1	1	1	0	0	0	0	0
VPN Certificate Objects MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-js-cert.txt	0	0	0	0	0	1	0	1	1	1
VPN MIB http://www.juniper.net/techpubs/en_US/junos14.1/topics/reference/mibs/mib-jnx-vpn.txt	1	1	1	1	1	1	0	0	0	0

Related Documentation

- [Juniper Networks Enterprise-Specific MIBs on page 34](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 63](#)
- [Standard SNMP MIBs Supported by Junos OS on page 15](#)
- [Loading MIB Files to a Network Management System on page 65](#)

SNMP MIB Objects Supported by Junos OS for the Set Operation

The following table lists the SNMP MIB objects that are supported for the **snmp set** operation by Junos OS.

Object Name	Object Identifier
RFC 1907	

Object Name	Object Identifier
sysContact	1.3.6.1.2.1.1.4
sysName	1.3.6.1.2.1.1.5
sysLocation	1.3.6.1.2.1.1.6
snmpEnableAuthenTraps	1.3.6.1.2.1.1.30
RFC 2819a	
alarmInterval	1.3.6.1.2.1.16.3.1.1.2
alarmVariable	1.3.6.1.2.1.16.3.1.1.2
alarmSampleType	1.3.6.1.2.1.16.3.1.1.4
alarmStartupAlarm	1.3.6.1.2.1.16.3.1.1.6
alarmRisingThreshold	1.3.6.1.2.1.16.3.1.1.7
alarmFallingThreshold	1.3.6.1.2.1.16.3.1.1.8
alarmRisingEventIndex	1.3.6.1.2.1.16.3.1.1.9
alarmFallingEventIndex	1.3.6.1.2.1.16.3.1.1.10
alarmOwner	1.3.6.1.2.1.16.3.1.1.11
alarmStatus	1.3.6.1.2.1.16.3.1.1.12
eventDescription	1.3.6.1.2.1.16.9.1.1.2
eventType	1.3.6.1.2.1.16.9.1.1.3
eventCommunity	1.3.6.1.2.1.16.9.1.1.4
eventOwner	1.3.6.1.2.1.16.9.1.1.6
eventStatus	1.3.6.1.2.1.16.9.1.1.7
RFC 2925a	
pingMaxConcurrentRequests	1.3.6.1.2.1.80.1.1
pingCtlTargetAddressType	1.3.6.1.2.1.80.1.2.1.3
pingCtlTargetAddress	1.3.6.1.2.1.80.1.2.1.4

Object Name	Object Identifier
pingCtlDataSize	1.3.6.1.2.1.80.1.2.1.5
pingCtlTimeOut	1.3.6.1.2.1.80.1.2.1.6
pingCtlProbeCount	1.3.6.1.2.1.80.1.2.1.7
pingCtlAdminStatus	1.3.6.1.2.1.80.1.2.1.8
pingCtlDataFill	1.3.6.1.2.1.80.1.2.1.9
pingCtlFrequency	1.3.6.1.2.1.80.1.2.1.10
pingCtlMaxRows	1.3.6.1.2.1.80.1.2.1.11
pingCtlStorageType	1.3.6.1.2.1.80.1.2.1.12
pingCtlTrapGeneration	1.3.6.1.2.1.80.1.2.1.13
pingCtlTrapProbeFailureFilter	1.3.6.1.2.1.80.1.2.1.14
pingCtlTrapTestFailureFilter	1.3.6.1.2.1.80.1.2.1.15
pingCtlType	1.3.6.1.2.1.80.1.2.1.16
pingCtlDescr	1.3.6.1.2.1.80.1.2.1.17
pingCtlSourceAddressType	1.3.6.1.2.1.80.1.2.1.18
pingCtlSourceAddress	1.3.6.1.2.1.80.1.2.1.19
pingCtlIfIndex	1.3.6.1.2.1.80.1.2.1.20
pingCtlByPassRouteTable	1.3.6.1.2.1.80.1.2.1.21
pingCtlDSField	1.3.6.1.2.1.80.1.2.1.22
pingCtlRowStatus	1.3.6.1.2.1.80.1.2.1.23
RFC 2925B	
traceRouteMaxConcurrentRequests	1.3.6.1.2.1.81.1.1
traceRouteCtlTargetAddressType	1.3.6.1.2.1.81.1.2.1.3
traceRouteCtlTargetAddress	1.3.6.1.2.1.81.1.2.1.4
traceRouteCtlByPassRouteTable	1.3.6.1.2.1.81.1.2.1.5

Object Name	Object Identifier
traceRouteCtlDataSize	1.3.6.1.2.1.81.1.2.1.6
traceRouteCtlTimeOut	1.3.6.1.2.1.81.1.2.1.7
traceRouteCtlProbesPerHop	1.3.6.1.2.1.81.1.2.1.8
traceRouteCtlPort	1.3.6.1.2.1.81.1.2.1.9
traceRouteCtlMaxTtl	1.3.6.1.2.1.81.1.2.1.10
traceRouteCtlDSField	1.3.6.1.2.1.81.1.2.1.11
traceRouteCtlSourceAddressType	1.3.6.1.2.1.81.1.2.1.12
traceRouteCtlSourceAddress	1.3.6.1.2.1.81.1.2.1.13
traceRouteCtlIfIndex	1.3.6.1.2.1.81.1.2.1.14
traceRouteCtlMiscOptions	1.3.6.1.2.1.81.1.2.1.15
traceRouteCtlMaxFailure	1.3.6.1.2.1.81.1.2.1.16
traceRouteCtlDontFragment	1.3.6.1.2.1.81.1.2.1.17
traceRouteCtlInitialTtl	1.3.6.1.2.1.81.1.2.1.18
traceRouteCtlFrequency	1.3.6.1.2.1.81.1.2.1.19
traceRouteCtlStorageType	1.3.6.1.2.1.81.1.2.1.20
traceRouteCtlAdminStatus	1.3.6.1.2.1.81.1.2.1.21
traceRouteCtlDescr	1.3.6.1.2.1.81.1.2.1.22
traceRouteCtlMaxRows	1.3.6.1.2.1.81.1.2.1.23
traceRouteCtlTrapGeneration	1.3.6.1.2.1.81.1.2.1.24
traceRouteCtlCreateHopEntries	1.3.6.1.2.1.81.1.2.1.25
traceRouteCtlType	1.3.6.1.2.1.81.1.2.1.26
traceRouteCtlRowStatus	1.3.6.1.2.1.81.1.2.1.27
Enterprise-Specific PING MIB	
jnxPingCtlIfName	1.3.6.1.4.1.2636.3.7.1.2.1.3

Object Name	Object Identifier
jnxPingCtlRoutingIfIndex	1.3.6.1.4.1.2636.3.7.1.2.1.4
jnxPingCtlRoutingIfName	1.3.6.1.4.1.2636.3.7.1.2.1.5
jnxPingCtlRoutingInstanceName	1.3.6.1.4.1.2636.3.7.1.2.1.6
jnxPingCtlRttThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.7
jnxPingCtlRttStdDevThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.8
jnxPingCtlRttJitterThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.9
jnxPingCtlEgressTimeThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.10
jnxPingCtlEgressStdDevThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.11
jnxPingCtlEgressJitterThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.12
jnxPingCtlIngressTimeThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.13
jnxPingCtlIngressStdDevThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.14
jnxPingCtlIngressJitterThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.15
jnxPingTrapGeneration	1.3.6.1.4.1.2636.3.7.1.2.1.16
Enterprise-Specific Traceroute MIB	
jnxTRCtlIfName	1.3.6.1.4.1.2636.3.8.1.2.1.3
jnxTRCtlRoutingInstanceName	1.3.6.1.4.1.2636.3.8.1.2.1.4
RFC 3413 Target MIB	
snmpTargetSpinLock	1.3.6.1.6.3.12.1.1
snmpTargetAddrTDomain	1.3.6.1.6.3.12.1.2.1.2
snmpTargetAddrTAddress	1.3.6.1.6.3.12.1.2.1.3
snmpTargetAddrTimeout	1.3.6.1.6.3.12.1.2.1.4
snmpTargetAddrRetryCount	1.3.6.1.6.3.12.1.2.1.5
snmpTargetAddrTagList	1.3.6.1.6.3.12.1.2.1.6
snmpTargetAddrParams	1.3.6.1.6.3.12.1.2.1.7

Object Name	Object Identifier
snmpTargetAddrStorageType	1.3.6.1.6.3.12.1.2.1.8
snmpTargetAddrRowStatus	1.3.6.1.6.3.12.1.2.1.9
snmpTargetParamsMPModel	1.3.6.1.6.3.12.1.3.1.2
snmpTargetParamsSecurityModel	1.3.6.1.6.3.12.1.3.1.3
snmpTargetParamsSecurityLevel	1.3.6.1.6.3.12.1.3.1.4
snmpTargetParamsSecurityName	1.3.6.1.6.3.12.1.3.1.5
snmpTargetParamsStorageType	1.3.6.1.6.3.12.1.3.1.6
snmpTargetParamsRowStatus	1.3.6.1.6.3.12.1.3.1.7
RFC 3413 Notify MIB	
snmpNotifyTag	1.3.6.1.6.3.13.1.1.1.2
snmpNotifyType	1.3.6.1.6.3.13.1.1.1.3
snmpNotifyStorageType	1.3.6.1.6.3.13.1.1.1.4
snmpNotifyRowStatus	1.3.6.1.6.3.13.1.1.1.5
snmpNotifyFilterProfileName	1.3.6.1.6.3.13.1.2.1.1
snmpNotifyFilterProfileStorType	1.3.6.1.6.3.13.1.2.1.2
snmpNotifyFilterProfileRowStatus	1.3.6.1.6.3.13.1.2.1.3
snmpNotifyFilterMask	1.3.6.1.6.3.13.1.3.1.2
snmpNotifyFilterType	1.3.6.1.6.3.13.1.3.1.3
snmpNotifyFilterStorageType	1.3.6.1.6.3.13.1.3.1.4
snmpNotifyFilterRowStatus	1.3.6.1.6.3.13.1.3.1.5
RFC 2574	
usmUserSpinLock	1.3.6.1.6.3.15.1.2.1
usmUserCloneFrom	1.3.6.1.6.3.15.1.2.2.1.4
usmUserAuthProtocol	1.3.6.1.6.3.15.1.2.2.1.5

Object Name	Object Identifier
usmUserAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.6
usmUserOwnAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.7
usmUserPrivProtocol	1.3.6.1.6.3.15.1.2.2.1.8
usmUserPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.9
usmUserOwnPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.10
usmUserPublic	1.3.6.1.6.3.15.1.2.2.1.11
usmUserStorageType	1.3.6.1.6.3.15.1.2.2.1.12
usmUserStatus	1.3.6.1.6.3.15.1.2.2.1.13
RFC 2575	
vacmGroupName	1.3.6.1.6.3.16.1.2.1.3
vacmSecurityToGroupStorageType	1.3.6.1.6.3.16.1.2.1.4
vacmSecurityToGroupStatus	1.3.6.1.6.3.16.1.2.1.5
vacmAccessContextMatch	1.3.6.1.6.3.16.1.4.1.4
vacmAccessReadViewName	1.3.6.1.6.3.16.1.4.1.5
vacmAccessWriteViewName	1.3.6.1.6.3.16.1.4.1.6
vacmAccessNotifyViewName	1.3.6.1.6.3.16.1.4.1.7
vacmAccessStorageType	1.3.6.1.6.3.16.1.4.1.8
vacmAccessStatus	1.3.6.1.6.3.16.1.4.1.9
vacmViewSpinLock	1.3.6.1.6.3.16.1.5.1
vacmViewTreeFamilyMask	1.3.6.1.6.3.16.1.5.2.1.3
vacmViewTreeFamilyType	1.3.6.1.6.3.16.1.5.2.1.4
vacmViewTreeFamilyStorageType	1.3.6.1.6.3.16.1.5.2.1.5
vacmViewTreeFamilyStatus	1.3.6.1.6.3.16.1.5.2.1.6
RFC 2576	

Object Name	Object Identifier
snmpCommunityName	1.3.6.1.6.3.18.1.1.1.2
snmpCommunitySecurityName	1.3.6.1.6.3.18.1.1.1.3
snmpCommunityContextEngineID	1.3.6.1.6.3.18.1.1.1.4
snmpCommunityContextName	1.3.6.1.6.3.18.1.1.1.5
snmpCommunityTransportTag	1.3.6.1.6.3.18.1.1.1.6
snmpCommunityStorageType	1.3.6.1.6.3.18.1.1.1.7
snmpCommunityStatus	1.3.6.1.6.3.18.1.1.1.8
RFC 2576	
snmpTargetAddrMask	1.3.6.1.6.3.18.1.2.1.1
snmpTargetAddrMMS	1.3.6.1.6.3.18.1.2.1.2

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 15](#)
- [Juniper Networks Enterprise-Specific MIBs on page 34](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 46](#)

Standard SNMP Traps Supported on Devices Running Junos OS

This topic provides pointers to the standard SNMP traps supported by Junos OS.



NOTE: For scalability reasons, the MPLS traps are generated by the ingress router only. For information about disabling the generation of MPLS traps, see the *Junos OS MPLS Applications Library for Routing Devices* document.

- [Standard SNMP Version 1 Traps](#)
- [Standard SNMP Version 2 Traps](#)
- [Standard SNMP Traps on EX Series Ethernet Switches](#)
- [Unsupported Standard SNMP Traps](#)

**Related
Documentation**

- [Juniper Networks Enterprise-Specific SNMP Traps on page 63](#)
- [Juniper Networks Enterprise-Specific MIBs on page 34](#)
- [Standard SNMP MIBs Supported by Junos OS on page 15](#)

- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 88](#)
- *Managing Traps and Informs*

Juniper Networks Enterprise-Specific SNMP Traps

This topic provides pointers to the enterprise-specific SNMP traps supported by the Junos OS.



NOTE: All enterprise-specific SNMP traps supported by the Junos OS can be sent in version 1, 2, and 3 formats.

- *Juniper Networks Enterprise-Specific SNMP Version 1 Traps*
- *Juniper Networks Enterprise-Specific SNMP Version 2 Traps*
- *Juniper Networks Enterprise-Specific BGP Traps*
- *Juniper Networks Enterprise-Specific LDP Traps*
- *Juniper Networks Enterprise-Specific License MIB Notifications*
- *Juniper Networks Enterprise-Specific MIMSTP Traps*
- *Juniper Networks Enterprise-Specific MPLS Traps*



NOTE: For scalability reasons, the MPLS traps are generated by the ingress router only. For information about disabling the generation of MPLS traps, see the *Junos OS MPLS Applications Library for Routing Devices*.

- *Juniper Networks Enterprise-Specific Traps on EX Series Switches*
- *Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers*

Related Documentation

- [Standard SNMP Traps Supported on Devices Running Junos OS on page 62](#)
- [Juniper Networks Enterprise-Specific MIBs on page 34](#)
- [Standard SNMP MIBs Supported by Junos OS on page 15](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 88](#)
- *Managing Traps and Informs*

CHAPTER 4

Loading MIB Files to a Network Management System

- Loading MIB Files to a Network Management System on page 65

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 Junos OS Enterprise MIBs index at http://www.juniper.net/techpubs/en_US/release-independent/junos/mibs/mibs.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 Enterprise MIBs index page (http://www.juniper.net/techpubs/en_US/release-independent/junos/mibs/mibs.html).
2. Click the **TAR** or **ZIP** link under the appropriate release heading to download the Junos MIB package for that release.
3. Decompress the file (**.tar** or **.zip**) using an appropriate utility.
4. 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`
5. 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.

6. 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
7. 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 on page 15](#)

- [Juniper Networks Enterprise-Specific MIBs on page 34](#)

CHAPTER 5

Configuring SNMP

- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuring the System Contact on a Device Running Junos OS on page 75](#)
- [Configuring the System Location for a Device Running Junos OS on page 76](#)
- [Configuring the System Description on a Device Running Junos OS on page 76](#)
- [Configuring SNMP Details on page 77](#)
- [Configuring the System Name on page 78](#)
- [Configuring the Commit Delay Timer on page 79](#)
- [Filtering Duplicate SNMP Requests on page 79](#)
- [Configuring SNMP Communities on page 80](#)
- [Examples: Configuring the SNMP Community String on page 83](#)
- [Adding a Group of Clients to an SNMP Community on page 83](#)
- [Configuring a Proxy SNMP Agent on page 85](#)
- [Configuring SNMP Traps on page 86](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 88](#)
- [Configuring SNMP Trap Options on page 88](#)
- [Configuring SNMP Trap Groups on page 92](#)
- [Example: Configuring SNMP Trap Groups on page 94](#)
- [Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 95](#)
- [Example: Configuring Secured Access List Checking on page 95](#)
- [Filtering Interface Information Out of SNMP Get and GetNext Output on page 96](#)
- [Configuring MIB Views on page 97](#)
- [Example: Ping Proxy MIB on page 98](#)

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.

```

[edit]
snmp {
  alarm-management {
    alarm-list-name list-name {
      alarm-id id {
        alarm-state state {
          description alarm-description;
          notification-id notification-id-of-alarm;
          resource-prefix alarm-resource-prefix;
          varbind-index varbind-index-in-alarm-varbind-list;
          varbind-subtree alarm-varbind-subtree;
          varbind-value alarm-varbind-value;
        }
      }
    }
  }
}
client-list client-list-name {
  ip-addresses;
}
community community-name {
  authorization authorization;
  client-list-name client-list-name;
  clients {
    address <restrict>;
  }
  logical-system (SNMP) 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;
    memory-trace;
    no-remote-trace;
    no-default-memory-trace;
}
trap-group group-name {
    categories {
        category;
    }
    destination-port port-number;
    routing-instance instance;
    logical-system (SNMP) logical-system-name;
    targets {
        address;
    }
    version (all | v1 | v2);
}
trap-options {
    agent-address outgoing-interface;
    source-address address;
    enterprise-oid;
    logical-system (SNMP) logical-system-name {
        routing-instance routing-instance-name {
            source-address address;
        }
    }
    routing-instance (SNMP) 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 (SNMP) 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 9](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)

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 9](#)
 - [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
 - [Complete SNMPv3 Configuration Statements on page 102](#)

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 73](#)
 - [Configuring the System Location for a Device Running Junos OS on page 76](#)
 - [Configuring the System Description on a Device Running Junos OS on page 76](#)
 - [Configuring the System Name on page 78](#)
 - [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 73](#)
- [Configuring the System Contact on a Device Running Junos OS on page 75](#)
- [Configuring the System Description on a Device Running Junos OS on page 76](#)
- [Configuring the System Name on page 78](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 73](#)
- [Configuring the System Contact on a Device Running Junos OS on page 75](#)
- [Configuring the System Location for a Device Running Junos OS on page 76](#)
- [Configuring the System Name on page 78](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

Configuring SNMP Details

You can use SNMP to store basic administrative details, such as a contact name and the location of the device. Your management system can then retrieve this information remotely, when you are troubleshooting an issue or performing an audit. In SNMP terminology, these are the `sysContact`, `sysDescription`, and `sysLocation` objects found within the system group of MIB-2 (as defined in RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*). You can set initial values directly in the Junos OS configuration for each system being managed by SNMP.

To set the system contact details:

1. Set the system contact details by including the **contact** statement at the **[edit snmp]** hierarchy level, or in an appropriate configuration group as shown here.

This administrative contact is placed into the MIB II **sysContact** object.

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

```
[edit groups global snmp]
user@host# set contact contact
```

For example:

```
[edit groups global snmp]
user@host# set contact "Enterprise Support, (650) 555-1234"
```

2. Configure a system description.

This string is placed into the MIB II **sysDescription** object. If the description contains spaces, enclose it in quotation marks (" ").

```
[edit groups global snmp]
user@host# set description description
```

For example:

```
[edit groups global snmp]
user@host# set description "M10i router with 8 FPCs"
```

3. Configure a system location.

This string is placed into the MIB II **sysLocation** object. If the location contains spaces, enclose it in quotation marks (" ").

To specify the system location:

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

[edit groups global snmp]
user@host# set location location
```

For example:

```
[edit groups global snmp]
user@host# set location "London Corporate Office, Lab 5, Row 11, Rack C"
```

4. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

5. Commit the configuration.

```
user@host# commit
```

6. To verify the configuration, enter the **show snmp mib walk system** operational-mode command.

The **show snmp mib walk system** command performs a MIB walk through of the system table (from MIB-2 as defined in RFC 1213). The SNMP agent in Junos OS responds by printing each row in the table and its associated value. You can use the same command to perform a MIB walk through any part of the MIB tree supported by the agent.

```
user@host> show snmp mib walk system
sysDescr.0    = M10i router with 8 FPCs
sysObjectID.0 = jnxProductNameM10i
sysUpTime.0   = 173676474
sysContact.0  = Enterprise Support, (650) 555-1234
sysName.0     = host
sysLocation.0 = London Corporate Office, Lab 5, Row 11, Rack C
sysServices.0 = 4
```

Related Documentation

- [Configuring SNMP Communities on page 80](#)
- [Configuring SNMP Traps on page 86](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)

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 73](#)
- [Configuring the System Contact on a Device Running Junos OS on page 75](#)
- [Configuring the System Location for a Device Running Junos OS on page 76](#)
- [Configuring the System Description on a Device Running Junos OS on page 76](#)

- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 *CLI User Guide*.

Related Documentation

- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 73](#)
- [Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 95](#)
- [Filtering Interface Information Out of SNMP Get and GetNext Output on page 96](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

Configuring SNMP Communities

Configuring the SNMP agent in Junos OS is a straightforward task that shares many familiar settings common to other managed devices in your network. For example, you need to configure Junos OS with an SNMP community string and a destination for traps. Community strings are administrative names that group collections of devices and the agents that are running on them together into common management domains. If a manager and an agent share the same community, they can communicate with each other. An SNMP community defines the level of authorization granted to its members, such as which MIB objects are available, which operations (read-only or read-write) are valid for those objects, and which SNMP clients are authorized, based on their source IP addresses.

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 create a read-only SNMP community:

1. Enter the SNMP community used in your network.

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

Community names must be unique.



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

[edit groups global]
user@host# set snmp **community name**

This example uses the standard name **public** to create a community that gives limited read-only access.

[edit groups global]
user@host# set snmp **community public**

2. Define the authorization level for the community.

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 97](#).

```
[edit groups global snmp community name]
user@host# set authorization authorization
```

This example confines the public community to read-only access. Any SNMP client (for example, an SNMP management system) that belongs to the public community can read MIB variables but cannot set (change) them.

```
[edit groups global snmp community public]
user@host# set authorization read-only
```

3. Define a list of clients in the community who are authorized to communicate with the SNMP agent in Junos OS.

The **clients** statement lists the IP addresses of the clients (community members) that are allowed to use this community. List the clients by IP address and prefix. Typically, the list includes the SNMP network management system in your network or the address of your management network. If no **clients** statement is present, all clients are allowed. For **address**, you must specify an IPv4 or IPv6 address, not a hostname.

```
[edit groups global snmp community name]
user@host# set clients address
```

The following statement defines the hosts in the 192.168.1.0/24 network as being authorized in the public community.

```
[edit groups global snmp community public]
user@host# set clients 192.168.1.0/24
```

4. Define the clients that are not authorized within the community by specifying their IP address, followed by the **restrict** statement.

```
[edit groups global snmp community name]
user@host# set clients address restrict
```

The following statement defines all other hosts as being restricted from the public community.

```
[edit groups global snmp community public]
user@host# set clients 0/0 restrict
```

5. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

6. Commit the configuration.

```
user@host# commit
```

To create a read-write SNMP community:

1. Enter the SNMP community used in your network.

```
[edit groups global]
user@host# set snmp community name
```

This example standard community string **private** to identify the community granted read-write access to the SNMP agent running on the device.

```
[edit groups global]
user@host# set snmp community private
```

2. Define the authorization level for the community.

```
[edit groups global snmp community name]
user@host# set authorization authorization
```

This example confines the public community to read-only access. Any SNMP client (for example, an SNMP management system) that belongs to the public community can read MIB variables but cannot set (change) them.

```
[edit groups global snmp community public]
user@host# set authorization read-write
```

3. Define a list of clients in the community who are authorized to make changes to the SNMP agent in Junos OS.

List the clients by IP address and prefix.

```
[edit groups global snmp community name]
user@host# set clients address
```

For example:

```
[edit groups global snmp community private]
user@host# set clients 192.168.1.15/24
user@host# set clients 192.168.1.18/24
```

4. Define the clients that are not authorized within the community by specifying their IP address, followed by the **restrict** statement.

```
[edit groups global snmp community name]
user@host# set clients address restrict
```

The following statement defines all other hosts as being restricted from the public community.

```
[edit groups global snmp community private]
user@host# set clients 0/0 restrict
```

5. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- [Adding a Group of Clients to an SNMP Community on page 83](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Examples: Configuring the SNMP Community String on page 83](#)

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 SNMP Communities on page 80](#)

Adding a Group of Clients to an SNMP Community

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

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

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

You can configure a prefix list at the **[edit policy options]** hierarchy level. Support for prefix lists in the SNMP community configuration enables you to use a single list to configure the SNMP and routing policies. For more information about the **prefix-list** statement, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.

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

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



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

The following example shows how to define a client list:

```
[edit]
snmp {
  client-list 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 a Proxy SNMP Agent

Starting with Release 12.3, Junos OS enables you to assign one of the devices in the network as a proxy SNMP agent through which the network management system (NMS) can query other devices in the network. When you configure a proxy, you can specify the names of devices to be managed through the proxy SNMP agent.

When the NMS queries the proxy SNMP agent, the NMS specifies the community name (for SNMPv1 and SNMPv2) or the context and security name (for SNMPv3) associated with the device from which it requires the information.



NOTE: If you have configured authentication and privacy methods and passwords for SNMPv3, those parameters are also specified in the query for SNMPv3 information.

To configure a proxy SNMP agent and specify devices to be managed by the proxy SNMP agent, you can include the following configuration statements at the `[edit snmp]` hierarchy level:

```
proxy proxy-name{
  device-name device-name;
  logical-system logical-system {
    routing-instance routing-instance;
  }
  routing-instance routing-instance;
  <version-v1 | version-v2c> {
    snmp-community community-name;
    no-default-comm-to-v3-config;
  }
  version-v3 {
    security-name security-name;
    context context-name;
  }
}
```

- The **proxy** statement enables you to specify a unique name for the proxy configuration.
- The **version-v1**, **version-v2c**, and **version-v3** statements enable you to specify the SNMP version.
- The **no-default-comm-to-v3-config** statement is an optional statement at the `[edit snmp proxy proxy-name <version-v1 | version-v2c>]` hierarchy level that when included in the configuration requires you to manually configure the statements at the `[edit snmp v3 snmp-community community-name]` and `[edit snmp v3 vacm]` hierarchy levels.

If the **no-default-comm-to-v3-config** statement is not included at the `[edit snmp proxy proxy-name <version-v1 | version-v2c>]` hierarchy level, the `[edit snmp v3`

`snmp-community community-name`] and `[edit snmp v3 vacm]` hierarchy level configurations are automatically initialized.

- The **logical-system** and **routing-instance** statements are optional statements that enable you to specify logical system and routing instance names if you want to create proxies for logical systems or routing instances on the device.



NOTE: The community and security configuration for the proxy should match the corresponding configuration on the device that is to be managed.



NOTE: Because the proxy SNMP agent does not have trap forwarding capabilities, the devices that are managed by the proxy SNMP agent send the traps directly to the network management system.

You can use the **show snmp proxy** operational mode command to view proxy details on a device. The **show snmp proxy** command returns the proxy names, device names, SNMP version, community/security, and context information.

Related
Documentation

- [proxy \(snmp\) on page 355](#)

Configuring SNMP Traps

Traps are unsolicited messages sent from an SNMP agent to remote network management systems or trap receivers. Many enterprises use SNMP traps as part of a fault-monitoring solution, in addition to system logging. In Junos OS, SNMP traps are not forwarded by default, so you must configure a trap-group if you wish to use SNMP traps.

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 name of the trap group is embedded in SNMP trap notification packets as one variable binding (varbind) known as the community name.

To configure an SNMP trap:

1. Create a single, consistent source address that Junos OS applies to all outgoing traps in your device.

A source address is useful, because although most Junos OS devices have a number of outbound interfaces, using one source address helps a remote NMS to associate the source of the traps with an individual device

```
[edit groups global snmp]
user@host# set trap-options source-address address
```

This example uses the IP address of the loopback interface (lo0) as the source address for all the SNMP traps that originate from the device.

```
[edit groups global snmp]
user@host# set trap-options source-address lo0
```

2. Create a trap group in which you can list the types of traps to be forwarded and the targets (addresses) of the receiving remote management systems.

```
[edit groups global snmp trap-group group-name]
user@host# set version (all | v1 | v2) targets address
```

This example creates a trap group called **managers**, allows SNMP version 2-formatted notifications (traps) to be sent to the host at address 192.168.1.15. This statement forwards all categories of traps.

```
[edit groups global snmp trap-group managers]
user@host# set version v2 targets 192.168.1.15
```

3. Define the specific subset of trap categories to be forwarded.

For a list of categories, see [“Configuring SNMP Trap Groups” on page 92](#).

```
[edit groups global snmp trap-group group-name]
user@host# set categories category
```

The following statement configures the standard MIB-II authentication failures on the agent (the device).

```
[edit groups global snmp trap-group managers]
user@host# set categories authentication
```

4. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

5. Commit the configuration.

```
user@host# commit
```

6. To verify the configuration, generate an authentication failure trap.

This means that the SNMP agent received a request with an unknown community. Other traps types can also be spoofed as well.

This feature enables you to trigger SNMP traps from routers and ensure that they are processed correctly within your existing network management infrastructure. This is also useful for testing and debugging SNMP behavior on the switch or NMS.

Using the **monitor traffic** command, you can verify that the trap is sent to the network management system.

```
user@host> request snmp spoof-trap spoof-trap authenticationFailure
Spoof-trap request result: trap sent successfully
```

Related Documentation

- [Adding a Group of Clients to an SNMP Community on page 83](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Examples: Configuring the SNMP Community String on page 83](#)

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 88](#)
- [Configuring SNMP Trap Groups on page 92](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 (SNMP)
  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 92](#).

This topic contains the following sections:

- [Configuring the Source Address for SNMP Traps on page 89](#)
- [Configuring the Agent Address for SNMP Traps on page 91](#)
- [Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps on page 91](#)

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 (SNMP) 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 not specified 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 88](#)
- [Configuring SNMP Trap Groups on page 92](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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 [“Standard SNMP Traps Supported on Devices Running Junos OS” on page 62](#) and [“Juniper Networks Enterprise-Specific SNMP Traps” on page 63](#).

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
- **services**—Services 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

To enable the **authenticationFailure** SNMP trap, you can configure either the **startup** or the **authentication** type of SNMP traps for the trap group:

- **startup** type of SNMP traps:

```
trap-group customer {
  categories {
    startup;
  }
}
```

- **authentication** type of SNMP traps:

```
trap-group customer {
  categories {
    authentication;
  }
}
```

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 \(SNMP\)](#).

Related Documentation

- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 88](#)
- [Configuring SNMP Trap Options on page 88](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Example: Configuring SNMP Trap Groups on page 94](#)

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 92](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 88](#)
- [Configuring SNMP Trap Options on page 88](#)

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 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Example: Configuring Secured Access List Checking on page 95](#)
- [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 95](#)
- [Filtering Interface Information Out of SNMP Get and GetNext Output on page 96](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

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;
  }
}
```

Starting with Release 12.1, Junos OS provides an except option (! operator) that enables you to filter out all interfaces except those interfaces that match all the regular expressions prefixed with the ! mark.

For example, to filter out all interfaces except the **ge** interfaces from the SNMP **get** and **get-next** results, enter the following command:

```
[edit snmp]
user@host# set filter-interfaces interfaces "!^~ge-.*"
user@host# commit
```

When this is configured, Junos OS filters out all interfaces except the **ge** interfaces from the SNMP **get** and **get-next** results.



NOTE: The ! mark is supported only as the first character of the regular expression. If it appears anywhere else in a regular expression, Junos OS considers the regular expression invalid, and returns an error.

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 95](#)
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

Configuring MIB Views

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

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

To create a MIB view and assign it to a community:

1. Configure a MIB view.

Although most network management systems use SNMPv3, Junos OS allows the use of MIB views with both SNMPv1 and SNMPv2c communities.

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.

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

```
[edit groups global snmp]
user@host# set view view-name oid object-identifier (include | exclude)
```

The following example creates a MIB view called ping-mib-view. The **oid** statement does not require a dot at the beginning of the object identifier. The **snmp view** statement includes the branch under the object identifier .1.3.6.1.2.1.80. This includes the entire DISMAN-PINGMIB subtree (as defined in RFC 2925, *Definitions of Managed Objects*

for Remote Ping, Traceroute, and Lookup Operations), which effectively permits access to any object under that branch.

```
[edit groups global snmp]
user@host# set view ping-mib-view oid 1.3.6.1.2.1.80 include
```

The following example adds a second branch in the same MIB view.

```
[edit groups global snmp]
user@host# set view ping-mib-view oid jnxPingMIB include
```

2. Assign a MIB view to a community that you want to control.

```
[edit groups global snmp community community-name]
user@host# set view view-name
```

This example creates a new community ping-mib which has read-write access to create entries within the DISMAN-PING-MIB.

```
[edit groups global snmp community ping-mib]
user@host# set authorization read-write
```

This example associate the MIB view created earlier with the new community.

```
[edit groups global snmp community ping-mib]
user@host# set view ping-mib-view
```

3. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

4. Commit the configuration.

```
user@host# commit
```

5. To verify, make sure that any member of the ping-mib community has read/write access to the branches that you specified under ping-mib-view.

Related Documentation

- *PING MIB*
- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Example: Ping Proxy MIB on page 98](#)
- *SNMP MIBs and Traps Reference*

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 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Configuring MIB Views on page 97](#)
- [view \(Configuring a MIB View\) on page 364](#)
- [oid on page 354](#)

CHAPTER 6

Configuring SNMPv3

- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: SNMPv3 Configuration on page 105](#)
- [Configuring the Local Engine ID on page 108](#)
- [Creating SNMPv3 Users on page 109](#)
- [Example: Creating SNMPv3 Users on page 110](#)
- [Configuring the SNMPv3 Authentication Type on page 111](#)
- [Configuring the SNMPv3 Encryption Type on page 112](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring the Access Privileges Granted to a Group on page 115](#)
- [Example: Configuring the Access Privileges Granted to a Group on page 118](#)
- [Assigning Security Model and Security Name to a Group on page 119](#)
- [Example: Configuring a Security Group on page 121](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 121](#)
- [Configuring the SNMPv3 Trap Notification on page 122](#)
- [Example: Configuring SNMPv3 Trap Notification on page 123](#)
- [Configuring the Trap Notification Filter on page 124](#)
- [Configuring the Trap Target Address on page 124](#)
- [Example: Configuring the Tag List on page 127](#)
- [Defining and Configuring the Trap Target Parameters on page 128](#)
- [Configuring SNMP Informs on page 131](#)
- [Configuring the Remote Engine and Remote User on page 132](#)
- [Example: Configuring the Remote Engine ID and Remote Users on page 133](#)
- [Configuring the Inform Notification Type and Target Address on page 136](#)
- [Example: Configuring the Inform Notification Type and Target Address on page 138](#)
- [Configuring the SNMPv3 Community on page 138](#)
- [Example: Configuring an SNMPv3 Community on page 140](#)

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-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 (SNMP) 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 109](#)
- [Configuring MIB Views on page 97](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 121](#)
- [Configuring SNMP Informs on page 131](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 109](#)
- [Configuring MIB Views on page 97](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 121](#)
- [Configuring SNMP Informs on page 131](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Example: SNMPv3 Configuration on page 105](#)

Example: SNMPv3 Configuration

Define an SNMPv3 configuration:

```

[edit snmp]
engine-id {
  use-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 102](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 the management port if the device has only one Routing Engine.

- Related Documentation**
- [Complete SNMPv3 Configuration Statements on page 102](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
 - [Example: SNMPv3 Configuration on page 105](#)

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 102](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
 - [Example: Creating SNMPv3 Users on page 110](#)
 - [Example: SNMPv3 Configuration on page 105](#)

Example: Creating SNMPv3 Users

Define SNMPv3 users:

```
[edit]
snmp {
  v3 {
    usm {
      local-engine {
        user user1 {
          authentication-md5 {
            authentication-password authentication-password;
          }
          privacy-des {
            privacy-password password;
          }
        }
        user user2 {
          authentication-sha {
            authentication-password authentication-password;
          }
          privacy-none;
        }
        user user3 {
          authentication-none;
          privacy-none;
        }
        user user4 {
          authentication-md5 {
            authentication-password authentication-password;
          }
          privacy-des {
            privacy-password authentication-password;
          }
        }
        user user5 {
          authentication-sha {
            authentication-password authentication-password;
          }
          privacy-aes128 {
            privacy-password authentication-password;
          }
        }
      }
    }
  }
}
```

- Related Documentation**
- [Complete SNMPv3 Configuration Statements on page 102](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 111](#)
- [Configuring SHA Authentication on page 111](#)
- [Configuring No Authentication on page 112](#)

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 SNMPv3 Encryption Type on page 112](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring the Access Privileges Granted to a Group on page 115](#)
- [Assigning Security Model and Security Name to a Group on page 119](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

Configuring the SNMPv3 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 112](#)
- [Configuring the Data Encryption Algorithm on page 113](#)
- [Configuring Triple DES on page 113](#)
- [Configuring No Encryption on page 113](#)

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 111](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Configuring the Access Privileges Granted to a Group on page 115](#)
- [Assigning Security Model and Security Name to a Group on page 119](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 97](#).

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 111](#)
- [Configuring the Access Privileges Granted to a Group on page 115](#)
- [Assigning Security Model and Security Name to a Group on page 119](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

Configuring the Access Privileges Granted to a Group

This topic includes the following sections:

- [Configuring the Group on page 115](#)
- [Configuring the Security Model on page 116](#)
- [Configuring the Security Level on page 116](#)
- [Associating MIB Views with an SNMP User Group on page 116](#)

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 97](#).

This section describes the following topics related to this configuration:

- [Configuring the Notify View on page 117](#)
- [Configuring the Read View on page 117](#)
- [Configuring the Write View on page 118](#)

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 111](#)
- [Defining Access Privileges for an SNMP Group on page 114](#)
- [Assigning Security Model and Security Name to a Group on page 119](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring the Access Privileges Granted to a Group on page 118](#)

Example: Configuring the Access Privileges Granted to a Group

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 wv2;
      }
    }
  }
}
group group3 {
  default-context-prefix {
    security-model v1 {      #Define an SNMPv3 security model
      security-level none {
        read-view rv3;
        write-view wv3;
      }
    }
  }
}

```

Related Documentation

- [Configuring the Access Privileges Granted to a Group on page 115](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 119](#)
- [Assigning Security Names to Groups on page 120](#)
- [Configuring the Group on page 120](#)

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 109](#). For information about configuring a community string, see [“Configuring the SNMPv3 Community” on page 138](#).



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 114](#).

Example: Configuring a Security Group

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 119](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 131](#).

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



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

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

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

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

**Related
Documentation**

- [Configuring the SNMPv3 Trap Notification on page 122](#)
- [Configuring the Trap Notification Filter on page 124](#)
- [Configuring the Trap Target Address on page 124](#)
- [Defining and Configuring the Trap Target Parameters on page 128](#)
- [Configuring SNMP Informs on page 131](#)
- [Configuring the Remote Engine and Remote User on page 132](#)
- [Configuring the Inform Notification Type and Target Address on page 136](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 126](#).

Related Documentation

- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 121](#)
- [Configuring the Trap Notification Filter on page 124](#)
- [Configuring the Trap Target Address on page 124](#)
- [Defining and Configuring the Trap Target Parameters on page 128](#)
- [Configuring SNMP Informs on page 131](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring SNMPv3 Trap Notification on page 123](#)

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 121](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 121](#)
- [Configuring the SNMPv3 Trap Notification on page 122](#)
- [Configuring the Trap Target Address on page 124](#)
- [Defining and Configuring the Trap Target Parameters on page 128](#)
- [Configuring SNMP Informs on page 131](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 (SNMP) 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 125](#)
- [Configuring the Address Mask on page 125](#)
- [Configuring the Port on page 126](#)
- [Configuring the Routing Instance on page 126](#)
- [Configuring the Trap Target Address on page 126](#)
- [Applying Target Parameters on page 127](#)

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 138](#).

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 127](#).

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 122](#).



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 121](#)
- [Configuring the SNMPv3 Trap Notification on page 122](#)
- [Configuring the Trap Notification Filter on page 124](#)
- [Defining and Configuring the Trap Target Parameters on page 128](#)
- [Configuring SNMP Informs on page 131](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring the Tag List on page 127](#)

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 121](#)
- [Configuring the Trap Target Address on page 124](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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 129](#)
- [Configuring the Target Parameters on page 129](#)

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 124.

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 129](#)
- [Configuring the Security Model on page 130](#)
- [Configuring the Security Level on page 130](#)
- [Configuring the Security Name on page 130](#)

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 121](#)
- [Configuring the SNMPv3 Trap Notification on page 122](#)
- [Configuring the Trap Notification Filter on page 124](#)
- [Configuring the Trap Target Address on page 124](#)
- [Configuring SNMP Informs on page 131](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

Configuring SNMP Informs

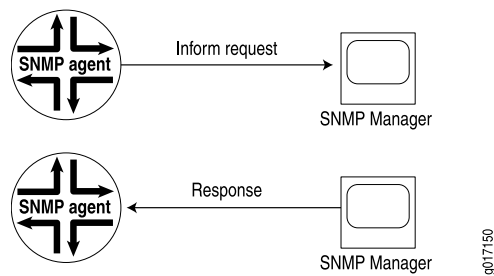
Junos OS supports two types of notifications: traps and informs. With traps, the receiver does not send any acknowledgment when it receives a trap. Therefore, the sender cannot determine if the trap was received. A trap may be lost because a problem occurred during transmission. To increase reliability, 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 132](#)). 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 121.

Related Documentation

- [Configuring SNMPv3 Traps on a Device Running Junos OS](#) on page 121
- [Configuring the Remote Engine and Remote User](#) on page 132
- [Configuring the Inform Notification Type and Target Address](#) on page 136
- [Complete SNMPv3 Configuration Statements](#) on page 102
- [Minimum SNMPv3 Configuration on a Device Running Junos OS](#) on page 104

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 121](#)
- [Configuring SNMP Informs on page 131](#)
- [Configuring the Inform Notification Type and Target Address on page 136](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring the Remote Engine ID and Remote Users on page 133](#)

Example: Configuring the Remote Engine ID and Remote Users

This example shows how to configure a remote engine and remote user to receive and respond to SNMP informs.

- [Requirements on page 133](#)
- [Overview on page 134](#)
- [Configuration on page 134](#)
- [Verification on page 135](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

This feature requires the use of plain-text passwords valid for SNMPv3. 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.

Though quotation marks are not always required to enclose passwords, it is a good idea to use them. You need them if the password has any spaces or possibly in the case of certain special characters or punctuation.

Overview

Inform notifications have the advantages that they can be authenticated and encrypted. They are also more reliable than traps. An SNMP manager that receives an inform acknowledges the message with a response.

Unlike traps, informs are stored and retransmitted at regular intervals until one of these conditions occurs:

- The target of the inform returns an acknowledgment to the SNMP agent
- A specified number of unsuccessful retransmissions have been attempted

Informs are supported in SNMPv3 to increase reliability.

For inform notifications, the remote engine ID identifies the SNMP agent on the remote device where the user resides, and the username identifies the user on a remote SNMP engine who receives the informs.

Configuration

Consider a scenario in which you have the following values to use in configuring the remote engine ID and remote user in this example:

Table 6: Values to Use in Example

Name of Variable	Value
username	u10
remote engine ID	800007E5804089071BC6D10A41
authentication type	authentication-md5
authentication password	qol67R%?
encryption type	privacy-des
privacy password	m*72Jl9v

- [Configuring Remote Engine and Remote User on page 135](#)
- [Results on page 135](#)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit snmp v3]** hierarchy level:

```
set usm remote-engine 800007E5804089071BC6D10A41 user u10 authentication-md5
authentication-key "qol67R%?"
set usm remote-engine 800007E5804089071BC6D10A41 user u10 privacy-des privacy-key
"m*72Jl9v"
```

Configuring Remote Engine and Remote User

Step-by-Step Procedure The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure the Remote Engine ID and remote user:

1. Configure the remote engine ID, username, and authentication type and password.

```
[edit snmp v3]
user@host# set usm remote-engine 800007E5804089071BC6D10A41 user u10
authentication-md5 authentication-key "qol67R%?"
```

2. Configure the encryption type and privacy password. You can configure only one encryption type per SNMPv3 user.

```
[edit snmp v3]
user@host# set usm remote-engine 800007E5804089071BC6D10A41 user u10
privacy-des privacy-key "m*72Jl9v"
```

Results

In configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit snmp v3]
user@ host# show
usm {
  remote-engine 800007E5804089071BC6D10A41 {
    user u10 {
      authentication-md5 {
        authentication-key "$9$Tz/teK8NdsLXk.f5n6p0ORev"; ## SECRET-DATA
      }
      privacy-des {
        privacy-key "$9$/gyNCu1KvWdwYMWw2gJHkRhcrWx"; ## SECRET-DATA
      }
    }
  }
}
```

After you have confirmed that the interfaces are configured, enter the **commit** command in configuration mode.

Verification

Verifying the Configuration of the Remote Engine ID and Username

Purpose Verify the status of the engine ID and user information.

Action Display information about SNMP v3 engine ID and user.

```
user@host> show snmp v3
Local engine ID: 80 00 0a 4c 01 0a ff 03 e3
Engine boots:      3
Engine time:       769187 seconds
Max msg size:      65507 bytes

Engine ID: 80 00 07 e5 80 40 89 07 1b c6 d1 0a 41
  User                               Auth/Priv  Storage  Status
  u10                                md5/des   nonvolatile active
```

Meaning The output displays the following information:

- Local engine ID and detail about the engine
- Remote engine ID
- Username
- Authentication type and encryption (privacy) type that is configured for the user
- Type of storage for username, either volatile (configuration saved) or volatile (not saved)
- Status of the new user. Only users with an active status can use SNMPv3.

For more information about the **show snmp v3** operational command, see *show snmp v3* in the Network Management Configuration Guide.

- Related Documentation**
- [Configuring the SNMPv3 Encryption Type on page 112](#)
 - [Configuring the SNMPv3 Authentication Type on page 111](#)
 - [Configuring SNMP Informs on page 131](#)
 - [Configuring the Remote Engine and Remote User on page 132](#)

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 (SNMP) 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 126](#).

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 121](#)
- [Configuring SNMP Informs on page 131](#)
- [Configuring the Remote Engine and Remote User on page 132](#)

- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring the Inform Notification Type and Target Address on page 138](#)

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 136](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

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;
context context-name;
security-name security-name;
tag tag-name;
```

This section includes the following topics:

- [Configuring the Community Name on page 139](#)
- [Configuring the Context on page 139](#)
- [Configuring the Security Names on page 140](#)
- [Configuring the Tag on page 140](#)

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 Context

An SNMP context defines a collection of management information that is accessible to an SNMP entity. Typically, an SNMP entity has access to multiple contexts. A context

can be a physical or logical system, a collection of multiple systems, or even a subset of a system. Each context in a management domain has a unique identifier.

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

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



NOTE: To query a routing instance or a logical system,

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 109](#)
- [Complete SNMPv3 Configuration Statements on page 102](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)
- [Example: Configuring an SNMPv3 Community on page 140](#)

Example: Configuring an SNMPv3 Community

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 138](#)
 - [Complete SNMPv3 Configuration Statements on page 102](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 104](#)

CHAPTER 7

Configuring SNMP for Routing Instances

- [Understanding SNMP Support for Routing Instances on page 143](#)
- [SNMP MIBs Supported for Routing Instances on page 144](#)
- [Support Classes for MIB Objects on page 154](#)
- [SNMP Traps Supported for Routing Instances on page 155](#)
- [Identifying a Routing Instance on page 156](#)
- [Enabling SNMP Access over Routing Instances on page 157](#)
- [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)
- [Example: Configuring Interface Settings for a Routing Instance on page 158](#)
- [Configuring Access Lists for SNMP Access over Routing Instances on page 159](#)

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.

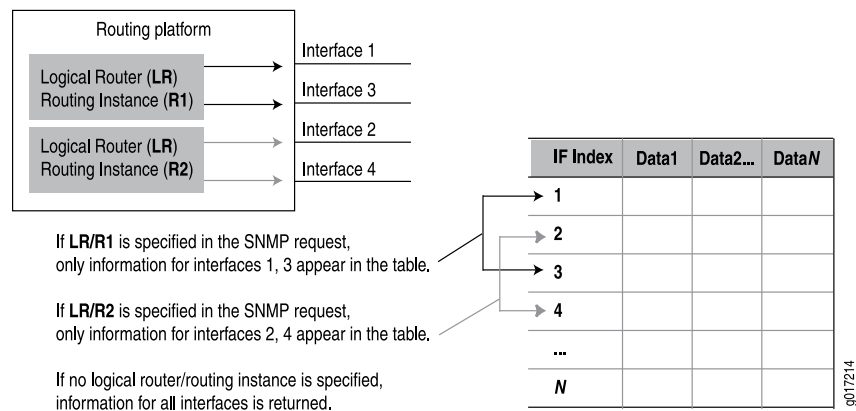
In Junos OS:

- 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 144](#)). 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 154](#)
- [SNMP Traps Supported for Routing Instances on page 155](#)
- [Identifying a Routing Instance on page 156](#)
- [Enabling SNMP Access over Routing Instances on page 157](#)
- [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)
- [Configuring Access Lists for SNMP Access over Routing Instances on page 159](#)

SNMP MIBs Supported for Routing Instances

Table 7 on page 144 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 7: MIB Support for Routing Instances (Juniper Networks MIBs)

Object	Support Class	Description/Notes
jnxProducts(1)	–	Product Object IDs
jnxServices(2)	–	Services

Table 7: MIB Support for Routing Instances (Juniper Networks MIBs) (continued)

Object	Support Class	Description/Notes
jnxMibs(3) jnxBoxAnatomy(1)	Class 3	Objects are exposed only for the default logical system.
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 7: 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 7: 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 8 on page 148](#) 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 8: 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 8: 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 8: 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 8: 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 9 on page 152 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 9: 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 NOTE: The igmpmib.mib is the draft version of the IGMP Standard MIB in the experimental tree. Junos OS does not support the original IGMP Standard MIB.
	l3vpn.mib	mplsVpnMIB
	jnx-mpls.mib	Example: mplsLspList
	jnx-ldp.mib	jnxLdp Example: jnxLdpStatsTable
	jnx-vpn.mib	jnxVpnMIB
	jnx-bgp.mib	jnxBgpMIB
	jnx-bgp-mib2.mib	jnxBgpM2Experiment

Table 10 on page 153 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 10: 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 11 on page 154 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 11: 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 143](#)
 - [Support Classes for MIB Objects on page 154](#)
 - [SNMP Traps Supported for Routing Instances on page 155](#)

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 **sysApplMIB**), 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 “[SNMP MIBs Supported for Routing Instances](#)” on page 144 for a list of the objects associated with each class.

**Related
Documentation**

- [Understanding SNMP Support for Routing Instances on page 143](#)
- [SNMP Traps Supported for Routing Instances on page 155](#)

SNMP Traps Supported 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 143](#)
- [Support Classes for MIB Objects on page 154](#)
- [SNMP MIBs Supported for Routing Instances on page 144](#)

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 143](#)

- [Enabling SNMP Access over Routing Instances on page 157](#)
- [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)

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 143](#)
- [Identifying a Routing Instance on page 156](#)
- [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)
- [Configuring Access Lists for SNMP Access over Routing Instances on page 159](#)

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 143](#)
- [Identifying a Routing Instance on page 156](#)
- [Enabling SNMP Access over Routing Instances on page 157](#)
- [Configuring Access Lists for SNMP Access over Routing Instances on page 159](#)
- [Example: Configuring Interface Settings for a Routing Instance on page 158](#)

Example: Configuring Interface Settings for a Routing Instance

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

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

```

router# snmpwalk -Os router1 INFrt@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 143](#)
 - [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)

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 143](#)
- [Enabling SNMP Access over Routing Instances on page 157](#)
- [Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157](#)

CHAPTER 8

Configuring SNMP Remote Operations

- [SNMP Remote Operations Overview on page 161](#)
- [Using the Ping MIB for SNMP Remote Operations on page 164](#)
- [Starting a Ping Test on page 164](#)
- [Monitoring a Running Ping Test on page 166](#)
- [Gathering Ping Test Results on page 169](#)
- [Stopping a Ping Test on page 170](#)
- [Interpreting Ping Variables on page 170](#)
- [Using the Traceroute MIB for SNMP Remote Operations on page 171](#)
- [Starting a Traceroute Test on page 172](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Stopping a Traceroute Test on page 179](#)
- [Interpreting Traceroute Variables on page 180](#)

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 *SNMP MIBs and Traps Reference*.

This topic covers the following sections:

- [SNMP Remote Operation Requirements on page 162](#)
- [Setting SNMP Views on page 162](#)
- [Setting Trap Notification for Remote Operations on page 163](#)
- [Using Variable-Length String Indexes on page 163](#)
- [Enabling Logging on page 164](#)

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 SNMP Communities” on page 80](#) and [community \(SNMP\)](#).

For more information about the **view** statement, see [“Configuring MIB Views” on page 97](#), [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 92](#).

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 181.

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 SNMP Remote Operations on page 164](#)
- [Using the Traceroute MIB for SNMP Remote Operations on page 171](#)

Using the Ping MIB for SNMP Remote Operations

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 161](#)
- [Starting a Ping Test on page 164](#)
- [Monitoring a Running Ping Test on page 166](#)
- [Gathering Ping Test Results on page 169](#)
- [Stopping a Ping Test on page 170](#)
- [Interpreting Ping Variables on page 170](#)

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 162](#).

There are two ways to start a ping test:

- [Using Multiple Set Protocol Data Units \(PDUs\) on page 165](#)
- [Using a Single Set PDU on page 165](#)

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 166](#)
- [pingProbeHistoryTable on page 167](#)
- [Generating Traps on page 168](#)

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 167.

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 92](#) and [“Example: Setting Trap Notification for Remote Operations” on page 163](#).

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 166](#). For more information about Ping MIB traps, see [“Generating Traps” on page 168](#).

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 12 on page 169](#).

Table 12: 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 13 on page 170](#).

Table 13: 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 14 on page 170](#).

Table 14: 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 SNMP Remote Operations

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 161](#)
- [Starting a Traceroute Test on page 172](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Stopping a Traceroute Test on page 179](#)
- [Interpreting Traceroute Variables on page 180](#)

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 162](#).

There are two ways to start a traceroute test:

- [Using Multiple Set PDUs on page 172](#)
- [Using a Single Set PDU on page 172](#)

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 SNMP Remote Operations on page 171](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [SNMP Remote Operations Overview on page 161](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Stopping a Traceroute Test on page 179](#)
- [Interpreting Traceroute Variables on page 180](#)

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 173](#)
- [traceRouteProbeResultsTable on page 174](#)
- [traceRouteHopsTable on page 175](#)
- [Generating Traps on page 176](#)

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 92](#) and [“Example: Setting Trap Notification for Remote Operations” on page 163](#).

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 SNMP Remote Operations on page 171](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [SNMP Remote Operations Overview on page 161](#)
- [Starting a Traceroute Test on page 172](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Stopping a Traceroute Test on page 179](#)
- [Interpreting Traceroute Variables on page 180](#)

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 173](#). For more information about Traceroute MIB traps, see the Generating Traps section in [“Monitoring a Running Traceroute Test” on page 173](#).

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 15 on page 178](#).

Table 15: traceRouteProbeHistoryTable

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

Table 15: traceRouteProbeHistoryTable (*continued*)

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

**Related
Documentation**

- [Using the Traceroute MIB for SNMP Remote Operations on page 171](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [SNMP Remote Operations Overview on page 161](#)
- [Starting a Traceroute Test on page 172](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Stopping a Traceroute Test on page 179](#)
- [Interpreting Traceroute Variables on page 180](#)

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 SNMP Remote Operations on page 171](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [SNMP Remote Operations Overview on page 161](#)
- [Starting a Traceroute Test on page 172](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Interpreting Traceroute Variables on page 180](#)

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 SNMP Remote Operations on page 171](#)
- [Monitoring a Running Traceroute Test on page 173](#)
- [SNMP Remote Operations Overview on page 161](#)
- [Starting a Traceroute Test on page 172](#)
- [Monitoring Traceroute Test Completion on page 177](#)
- [Gathering Traceroute Test Results on page 178](#)
- [Stopping a Traceroute Test on page 179](#)

CHAPTER 9

Tracing SNMP Activity

- [Tracing SNMP Activity on a Device Running Junos OS on page 181](#)
- [Example: Tracing SNMP Activity on page 184](#)

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;  
    memory-trace;  
    no-remote-trace;  
    no-default-memory-trace;  
}
```

These statements are described in the following sections:

- [Configuring the Number and Size of SNMP Log Files on page 182](#)
- [Configuring Access to the Log File on page 182](#)
- [Configuring a Regular Expression for Lines to Be Logged on page 183](#)
- [Configuring the Trace Operations on page 183](#)

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 16 on page 183 describes the meaning of the SNMP tracing flags.

Table 16: SNMP Tracing Flags

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

Table 16: SNMP Tracing Flags (*continued*)

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

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

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

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

Related Documentation

- [Configuring SNMP on a Device Running Junos OS on page 73](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)
- [Example: Tracing SNMP Activity on page 184](#)
- [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 73](#)
- [Tracing SNMP Activity on a Device Running Junos OS on page 181](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level on page 69](#)

CHAPTER 10

SNMP FAQs

- [Junos OS SNMP FAQs Overview on page 187](#)
- [Junos OS SNMP FAQs on page 188](#)

Junos OS SNMP FAQs Overview

SNMP enables users to monitor network devices from a central location. Many network management systems (NMS) are based on SNMP, and support for this protocol is a key feature of most network devices.

Juniper Networks provides many different platforms that support SNMP on Junos OS. Junos OS includes an onboard SNMP agent that provides remote management applications with access to detailed information about the devices on the network.

A typical SNMP implementation contains three components:

- Managed devices – Such as routers and switches.
- SNMP agent – Process that resides on a managed device and communicates with the NMS.
- NMS – A combination of hardware and software used to monitor and administer the network; network device that runs SNMP manager software. Also referred to as an SNMP manager.

The SNMP agent exchanges network management information with the SNMP manager (NMS). The agent responds to requests for information and actions from the manager. The SNMP manager collects information about network connectivity, activity, and events by polling managed devices.

SNMP implementation in Junos OS uses a master SNMP agent (known as an SNMP process or `snmpd`) that resides on the managed device. Various subagents reside on different modules of Junos OS as well (such as the Routing Engine), and these subagents are managed by the `snmpd`.

Related Documentation

- [Junos OS SNMP FAQs on page 188](#)

Junos OS SNMP FAQs

This Frequently Asked Questions technology overview covers these SNMP-related areas:

- [Junos OS SNMP Support FAQs on page 188](#)
- [Junos OS MIBs FAQs on page 189](#)
- [Junos OS SNMP Configuration FAQs on page 196](#)
- [SNMPv3 FAQs on page 200](#)
- [SNMP Interaction with Juniper Networks Devices FAQs on page 202](#)
- [SNMP Traps and Informs FAQs on page 204](#)
- [Junos OS Dual Routing Engine Configuration FAQs on page 210](#)
- [SNMP Support for Routing Instances FAQs on page 211](#)
- [SNMP Counters FAQs on page 212](#)

Junos OS SNMP Support FAQs

This section presents frequently asked questions and answers related to SNMP support on Junos OS.

Which SNMP versions does Junos OS support?

Junos OS supports SNMP version 1 (SNMPv1), version 2 (SNMPv2c), and version 3 (SNMPv3). By default, SNMP is disabled on a Juniper Networks device.

Which ports (sockets) does SNMP use?

The default port for SNMP queries is port 161. The default port for SNMP traps and informs is port 162. The ports used by SNMP are configurable, and you can configure your system to use ports other than the defaults.

Is SNMP support different among the Junos OS platforms?

No, SNMP support is not different among the Junos OS platforms. SNMP configuration, interaction, and behavior are the same on any Junos OS device. The only difference that might occur across platforms is MIB support.

See also the *SNMP MIBs and Traps Reference* for a list of MIBs that are supported across the Junos OS platforms.

Does Junos OS support the user-based security model (USM)?

Yes, Junos OS supports USM as part of its support for SNMPv3. SNMPv3 contains more security measures than previous versions of SNMP, including providing a defined USM. SNMPv3 USM provides message security through data integrity, data origin authentication, message replay protection, and protection against disclosure of the message payload.

Does Junos OS support the view-based access control model (VACM)?

Yes, Junos OS supports VACM as part of its support for SNMPv3. SNMPv3 contains more security measures than previous versions of SNMP, including providing a defined VACM. SNMPv3 VACM determines whether a specific type of access (read or write) to the management information is allowed.

Does Junos OS support SNMP informs?

Yes, Junos OS supports SNMP informs as part of its support for SNMPv3. SNMP informs are confirmed notifications sent from SNMP agents to SNMP managers when significant events occur on a network device. When an SNMP manager receives an inform, it sends a response to the sender to verify receipt of the inform.

Can I provision or configure a device using SNMP on Junos OS?

No, provisioning or configuring a device using SNMP is not allowed on Junos OS.

**Related
Documentation****Junos OS MIBs FAQs**

This section presents frequently asked questions and answers related to Junos OS MIBs.

What is a MIB?

A management information base (MIB) is a table of definitions for managed objects in a network device. MIBs are used by SNMP to maintain standard definitions of all of the components and their operating conditions within a network device. Each object in the MIB has an identifying code called an object identifier (OID).

MIBs are either standard or enterprise-specific. Standard MIBs are created by the Internet Engineering Task Force (IETF) and documented in various RFCs. Enterprise-specific MIBs are developed and supported by a specific equipment manufacturer.

For a list of supported standard MIBs, see “[Standard SNMP MIBs Supported by Junos OS](#)” on page 15.

For a list of Juniper Networks enterprise-specific MIBs, see “[Juniper Networks Enterprise-Specific MIBs](#)” on page 34.

Do MIB files reside on the Junos OS devices?

No, MIB files do not reside on the Junos OS devices. You must download the MIB files from the Juniper Networks Technical Publications page for the required Junos OS release: http://www.juniper.net/techpubs/en_US/release-independent/junos/mibs/mibs.html.

How do I compile and load the Junos OS MIBs onto an SNMP manager or NMS?

For your network management systems (NMSs) to identify and understand the MIB objects used by 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 names, IDs, and data types for the NMS.

You can download the Junos OS MIB package from the Enterprise-Specific MIBs and Traps section at

http://www.juniper.net/techpubs/en_US/release-independent/junos/mibs/mibs.html or <http://www.juniper.net/techpubs/software/junos/index.html>.

The Junos OS MIB package has two folders: **StandardMibs**, containing standard MIBs supported on Juniper Networks devices, and **JuniperMibs**, containing Juniper Networks enterprise-specific MIBs. You *must* have the required standard MIBs downloaded and decompressed before downloading any enterprise-specific MIBs. There might be dependencies that require a particular standard MIB to be present on the compiler before loading a particular enterprise-specific MIB.

The Junos OS MIB package is available in **.zip** and **.tar** formats. Download the format appropriate for your requirements.

Use the following steps to load MIB files for devices running Junos OS:

1. Navigate to the appropriate Juniper Networks software download page and locate the **Enterprise MIBs** link under the **Enterprise-Specific MIBs and Traps** section.



NOTE: Although the link is titled **Enterprise MIBs**, both standard MIBs and enterprise-specific MIBs are available for download from this location.

2. Click the **TAR** or **ZIP** link to download the Junos OS MIB package.
3. Decompress the file (**.tar** or **.zip**) using an appropriate utility.



NOTE: Some commonly used MIB compilers are preloaded with standard MIBs. You can skip Step 4 and Step 5 and proceed to Step 6 if you already have the standard MIBs loaded on your system.

4. Load the standard MIB files from the **StandardMibs** folder.

Load the files in the following order:

- 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
5. Load any 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 standard MIB to be present on the compiler before loading a particular enterprise-specific MIB. Dependencies are listed in the **IMPORT** section of the MIB file.

6. After loading the standard MIBs, load the Juniper Networks enterprise-specific SMI MIB, **mib-jnx-smi.txt**, and the following optional SMI MIBs based on your requirements:
 - **mib-jnx-exp.txt**—(Recommended) for Juniper Networks experimental MIB objects
 - **mib-jnx-js-smi.txt**—(Optional) for Juniper Security MIB tree objects
 - **mib-jnx-ex-smi.txt**—(Optional) for EX Series Ethernet Switches
7. Load any remaining desired enterprise-specific MIBs from the **JuniperMibs** folder.



TIP: While loading a MIB file, if the compiler returns an error message indicating that any of the objects are 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 are not loaded on the compiler, load the missing file or files first, then try to load the MIB file that failed.

The system might return an error if files are not loaded in a particular order.

What is SMI?

Structure of Management Information Version (SMI) is a subset of Abstract Syntax Notation One (ASN.1), which describes the structure of objects. SMI is the notation syntax, or “grammar”, that is the standard for writing MIBs.

Which versions of SMI does Junos OS support?

The Junos OS supports SMIV1 for SNMPv1 MIBs, and SMIV2 for SNMPv2c and enterprise MIBs.

Does Junos OS support MIB II?

Yes, Junos OS supports MIB II, the second version of the MIB standard.

The features of MIB II include:

- Additions that reflect new operational requirements.
- Backward compatibility with the original MIBs and SNMP.
- Improved support for multiprotocol entities.
- Improved readability.

Refer to the relevant release documentation for a list of MIBs that are supported. Go to <http://www.juniper.net/techpubs/software/junos/index.html>.

Are the same MIBs supported across all Juniper Networks devices?

There are some common MIBs supported by all the Junos OS devices, such as the Interface MIB (ifTable), System MIB, and Chassis MIB. Some MIBs are supported only by functionalities on specific platforms. For example, the Bridge MIB is supported on the EX Series Ethernet Switches and the SRX Series Services Gateways for the branch.

What is the system object identifier (SYSOID) of a device? How do I determine the SYSOID of my device?

The jnx-chas-defines (Chassis Definitions for Router Model) MIB has a **jnxProductName** branch for every Junos OS device. The system object ID of a device is identical to the object ID of the **jnxProductName** for the platform. For example, for an M7i Multiservice Edge Router, the jnxProductNameM7i is .1.3.6.1.4.1.2636.1.1.1.2.10 in the jnxProductName branch, which is identical to the SYSOID of the M7i (.1.3.6.1.4.1.2636.1.1.1.2.10).

How can I determine if a MIB is supported on a platform? How can I determine which MIBs are supported by a device?

MIBs device and platform support is listed on the Junos OS Technical Documentation index page. Go to <http://www.juniper.net/techpubs/software/junos/> and select your version or release of Junos OS. Navigate to the *SNMP MIBs and Traps Reference*. The *SNMP MIBs and Traps Reference* specifies which MIBs are supported on the different platforms.

What can I do if the MIB OID query is not responding?

There can be various reasons why the MIB OID query stops responding. One reason could be that the MIB itself is unresponsive. To verify that the MIB responds, use the **show snmp mib walk | get MIB name | MIB OID** command:

- If the MIB responds, the communication issue exists between the SNMP master and SNMP agent. Possible reasons for this issue include network issues, an incorrect community configuration, an incorrect SNMP configuration, and so on.
- If the MIB does not respond, enable SNMP **traceoptions** to log PDUs and errors. All incoming and outgoing SNMP PDUs are logged. Check the **traceoptions** output to see if there are any errors.

If you continue to have problems with the MIB OID query, technical product support is available through the Juniper Networks Technical Assistance Center (JTAC).

What is the enterprise branch number for Junos OS?

The enterprise branch number for Junos OS is 2636. Enterprise branch numbers are used in SNMP MIB configurations, and they are also known as SMI network management private enterprise codes.

Which MIB displays the hardware and chassis details on a Juniper Networks device?

The Chassis MIB (jnxchassis.mib) displays the hardware and chassis details for each Juniper Networks device. It provides information about the router and its components. The Chassis MIB objects represent each component and its status.

For more information about enterprise-specific Chassis MIBs, see *Chassis MIBs* in the *Junos OS SNMP MIBs and Traps Reference* document.

Does Junos OS support the Entity MIB?

No, Junos OS does not support the Entity MIB, which is designed to identify physical and logical elements of a managed device. Instead, Junos OS supports the enterprise-specific Chassis MIB to identify the chassis components on the device.

Which MIB objects can I query to determine the CPU and memory utilization of the Routing Engine, Flexible PIC Concentrator (FPC), and PIC components on a device?

Query the Chassis MIB objects `jnxOperatingMemory`, `jnxOperatingBuffer`, and `jnxOperatingCPU` to find out the CPU and memory utilization of the hardware components of a device.

Is the interface index (ifIndex) persistent?

For the Junos OS Release 10.0 and earlier, the ifIndex is persistent when reboots occur if the Junos OS version remains the same, meaning the values assigned to the interfaces in the ifIndex do not change. When there is a software upgrade, the device tries to keep the ifIndex persistent on a best effort basis.

For the Junos OS Release 10.1 and later, the ifIndex is persistent on all platforms, except for the EX4200 virtual chassis because it can have over 500 interfaces.

Is it possible to set the ifAdminStatus?

SNMP is not allowed to set the ifAdminStatus.

Which MIB objects support SNMP set operations?

The Junos OS SNMP set operations are supported in the following MIB tables and variables:

- `snmpCommunityTable`
- `eventTable`
- `alarmTable`
- `snmpTargetAddrExtTable`
- `jnxPingCtlTable`
- `pingCtlTable`
- `traceRouteCtlTable`
- `jnxTraceRouteCtlTable`
- `sysContact.O`

- sysName.0
- sysLocation.0
- pingMaxConcurrentRequests.0
- traceRouteMaxConcurrentRequests.0
- usmUserSpinLock
- usmUserOwnAuthKeyChange
- usmUserPublic
- vacmSecurityToGroupTable (vacmGroupName, vacmSecurityToGroupStorageType, and vacmSecurityToGroupStatus)
- vacmAccessTable (vacmAccessContextMatch, vacmAccessReadViewName, vacmAccessWriteViewName, vacmAccessNotifyViewName, vacmAccessStorageType, and vacmAccessStatus)
- vacmViewSpinLock
- vacmViewTreeFamilyTable (vacmViewTreeFamilyMask, vacmViewTreeFamilyType, vacmViewTreeFamilyStorageType, and vacmViewTreeFamilyStatus)

Does Junos OS support remote monitoring (RMON)?

Yes, Junos OS supports RMON as defined in RFC 2819, *Remote Network Monitoring Management Information Base*. However, remote monitoring version 2 (RMON 2) is not supported.

Can I use SNMP to determine the health of the processes running on the Routing Engine?

Yes, you can use SNMP to determine the health of the Routing Engine processes by configuring the health monitoring feature. On Juniper Networks devices, RMON alarms and events provide much of the infrastructure needed to reduce the polling overhead from the NMS. However, you must set up the NMS to configure specific MIB objects into RMON alarms. This often requires device-specific expertise and customizing the monitoring application. Additionally, some MIB object instances that need monitoring are set only at initialization, or they 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, such as file system usage, CPU usage, and memory usage, and includes support for unknown or dynamic object instances, such as Junos OS software processes.

To display the health monitoring configuration, use the **show snmp health-monitor** command:

```
user@host> show snmp health-monitor
interval 300;
rising-threshold 90;
falling-threshold 80;
```

When you configure the health monitor, monitoring information for certain object instances is available, as shown in [Table 17 on page 195](#).

Table 17: Monitored Object Instances

Object	Description
jnxHrStoragePercentUsed.1	Monitors the following file system on the router or switch: /dev/ad0s1a: This is the root file system mounted on /.
jnxHrStoragePercentUsed.2	Monitors the following file system on the router or switch: /dev/ad0s1e: This is the configuration file system mounted on /config.
jnxOperatingCPU (RE0)	Monitor CPU usage for Routing Engines RE0 and RE1. The index values assigned to the Routing Engines depend on whether the Chassis MIB uses a zero-based or a ones-based indexing scheme. Because the indexing scheme is configurable, the correct index is determined whenever the router 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.
jnxOperatingCPU (RE1)	
jnxOperatingBuffer (RE0)	Monitor the amount of memory available on Routing Engines RE0 and RE1. Because the indexing of this object is identical to that used for jnxOperatingCPU, index values are adjusted depending on the indexing scheme used in the Chassis MIB. As with jnxOperatingCPU, the alarm entry monitoring RE1 is removed if the router or switch has only one Routing Engine.
jnxOperatingBuffer (RE1)	
sysApplElmtRunCPU	Monitors the CPU usage for each Junos OS software process. Multiple instances of the same process are monitored and indexed separately.
sysApplElmtRunMemory	Monitors the memory usage for each Junos OS software process. Multiple instances of the same process are monitored and indexed separately.

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

Are the Ping MIBs returned in decimal notation and ASCII?

Yes, both decimal notation and ASCII are supported, which is the standard implementation in SNMP. All strings are ASCII encoded.

The following example displays the Ping MIB in hexadecimal notation:

```
pingCtlTargetAddress.2.69.72.9.116.99.112.115.97.109.112.108.101 = 0a fa 01 02
```

This translates to ASCII:

```
pingCtlTargetAddress."EH"."tcpsample" = 0a fa 01 02
2= length of the string
69=E
72=H
9=length of second string
116=t
```

```
99 =c
112=p
115=s
97=a
109=m
112 =p
108 =l
101 =e
```

As of Junos OS Release 9.6 and later, the Junos OS CLI returns ASCII values using the command **show snmp mib get | get-next | walk ascii**.

The following example shows the output with the ASCII option:

```
user@host> show snmp mib walk pingCtlTargetAddress ascii
pingCtlTargetAddress."EH"."httpgetsample" = http://www.yahoo.com
pingCtlTargetAddress."p1"."t2" = 74 c5 b3 06
pingCtlTargetAddress."p1"."t3" = 74 c5 b2 0c
```

The following example shows the output without the ASCII option:

```
user@host> show snmp mib walk pingCtlTargetAddress
pingCtlTargetAddress.2.69.72.13.104.116.116.112.103.101.116.115.97.109.112.108.101
= http://www.yahoo.com
pingCtlTargetAddress.2.112.49.2.116.50 = 74 c5 b3 06
pingCtlTargetAddress.2.112.49.2.116.51 = 74 c5 b2 0c
```

You can convert decimal and ASCII values using a decimal ASCII chart like the one at <http://www.asciichart.com>.

Is IPv6 supported by the Ping MIB for remote operations?

No, IPv6 is not supported.

Is there an SNMP MIB to show Address Resolution Protocol (ARP) table information? Are both IP and MAC addresses displayed in the same table?

Yes, the Junos OS supports the standard MIB **ipNetToMediaTable**, which is described in RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*. This table is used for mapping IP addresses to their corresponding MAC addresses.

Related Documentation

Junos OS SNMP Configuration FAQs

This section presents frequently asked questions and answers related to Junos OS SNMP configuration.

Can the Junos OS be configured for SNMPv1 and SNMPv3 simultaneously?

Yes, SNMP has backward compatibility, meaning that all three versions can be enabled simultaneously.

Can I filter specific SNMP queries on a device?

Yes, you can filter specific SNMP queries on a device using **exclude** and **include** statements.

The following example shows a configuration that blocks read-write operation on all OIDs under .1.3.6.1.2.1.1 for the community **test**:

```
user@host# show snmp
view system-exclude {
  oid .1.3.6.1.2.1.1 exclude;
  oid .1 include;
}
community test {
  view system-exclude;
  authorization read-write;
}
```

Can I change the SNMP agent engine ID?

Yes, the SNMP agent engine ID can be changed to the MAC address of the device, the IP address of the device, or any other desired value. Several examples are included here.

The following example shows how to use the MAC address of a device as the SNMP agent engine ID:

```
user@host# show snmp
engine-id {
  use-mac-address;
}
```

The following example shows how to use the IP address of a device as the SNMP agent engine ID:

```
user@host# show snmp
engine-id {
  use-default-ip-address;
}
```

The following example shows the use of a selected value, **AA** in this case, as the SNMP agent engine ID of a device:

```
user@host# show snmp
engine-id {
  local AA;
}
```

How can I configure a device with dual Routing Engines or a chassis cluster (for SRX Series Services Gateways or J Series Service Routers) for continued communication during a switchover?

When configuring for continued communication, the SNMP configuration should be identical between the Routing Engines. However, it is best to have separate Routing Engine IDs configured for each Routing Engine, especially when using SNMPv3.

The following example shows the configuration of the Routing Engines in a dual Routing Engine device. Notice that the Routing Engine IDs are set to the MAC addresses for each Routing Engine:

```
user@host# show groups
re0 {
  system {
    host-name PE3-re0;
```

```
}
interfaces {
  fxp0 {
    unit 0 {
      family inet {
        address 116.197.178.14/27;
        address 116.197.178.29/27 {
          master-only;
        }
      }
    }
  }
}
snmp {
  engine-id {
    use-mac-address;
  }
}
}
rel {
  system {
    host-name PE3-rel;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 116.197.178.11/27;
          address 116.197.178.29/27 {
            master-only;
          }
        }
      }
    }
  }
}
snmp {
  engine-id {
    use-mac-address;
  }
}
}
```

The following is an example of an SNMPv3 configuration on a dual Routing Engine device:

```
user@host> show snmp name host1
v3 {
  vacm {
    security-to-group {
      security-model usm {
        security-name test123 {
          group test1;
        }
        security-name juniper {
          group test1;
        }
      }
    }
  }
}
```

```

    }
    access {
        group test1 {
            default-context-prefix {
                security-model any {
                    security-level authentication {
                        read-view all;
                    }
                }
            }
            context-prefix MGMT_10 {
                security-model any {
                    security-level authentication {
                        read-view all;
                    }
                }
            }
        }
    }
}
target-address server1 {
    address 116.197.178.20;
    tag-list router1;
    routing-instance MGMT_10;
    target-parameters test;
}
target-parameters test {
    parameters {
        message-processing-model v3;
        security-model usm;
        security-level authentication;
        security-name juniper;
    }
    notify-filter filter1;
}
notify server {
    type trap;
    tag router1;
}
notify-filter filter1 {
    oid .1 include;
}
view all {
    oid .1 include;
}
community public {
    view all;
}
community comm1;
community comm2;
community comm3 {
    view all;
    authorization read-only;
    logical-system LDP-VPLS {
        routing-instance vpls-server1;
    }
}

```

```
}
trap-group server1 {
  targets {
    116.197.179.22;
  }
}
routing-instance-access;
traceoptions {
  flag all;
}
}
```

How can I track SNMP activities?

SNMP trace operations track activity of SNMP agents and record the information in log files.

A sample **traceoptions** configuration might look like this:

```
[edit snmp]
user@host# set traceoptions flag all
```

When the **traceoptions flag all** statement is included at the **[edit snmp]** hierarchy level, the following log files are created:

- snmpd
- mib2d
- rmopd

Related Documentation

- [Junos OS SNMP Support FAQs on page 188](#)
- [Junos OS MIBs FAQs on page 189](#)
- [SNMPv3 FAQs on page 200](#)
- [SNMP Interaction with Juniper Networks Devices FAQs on page 202](#)
- [SNMP Traps and Informs FAQs on page 204](#)
- [SNMP Support for Routing Instances FAQs on page 211](#)
- [SNMP Counters FAQs on page 212](#)

SNMPv3 FAQs

This section presents frequently asked questions and answers related to SNMPv3.

Why is SNMPv3 important?

SNMP v3 provides enhanced security compared to the other versions of SNMP. It provides authentication and encryption of data. Enhanced security is important for managing devices at remote sites from the management stations.

In my system, the MIB object snmpEngineBoots is not in sync between two Routing Engines in a dual Routing Engine device. Is this normal behavior?

Yes, this is the expected behavior. Each Routing Engine runs its own SNMP process (snmpd), allowing each Routing Engine to maintain its own engine boots. However, if both routing engines have the same engine ID and the routing engine with lesser **snmpEngineBoots** value is selected as the master routing engine during the switchover process, the **snmpEngineBoots** value of the master routing engine is synchronized with the **snmpEngineBoots** value of the other routing engine.

Do I need the SNMP manager engine object identifier (OID) for informs?

Yes, the engine OID of the SNMP manager is required for authentication, and informs do not work without it.

I see the configuration of informs under the [edit snmp v3] hierarchy. Does this mean I cannot use informs with SNMPv2c?

Informs can be used with SNMPv2c. The following example shows the basic configuration for SNMPv3 informs on a device (note that the authentication and privacy is set to none):

```
[edit snmp]
v3 {
  usm {
    remote-engine 00000063000100a2c0a845b3 {
      user RU2_v3_sha_none {
        authentication-none;
        privacy-none;
      }
    }
  }
  vacm {
    security-to-group {
      security-model usm {
        security-name RU2_v3_sha_none {
          group g1_usm_auth;
        }
      }
    }
  }
  access {
    group g1_usm_auth {
      default-context-prefix {
        security-model usm {
          security-level authentication {
            read-view all;
            write-view all;
            notify-view all;
          }
        }
      }
    }
  }
}
target-address TA2_v3_sha_none {
  address 192.168.69.179;
  tag-list tl1;
  address-mask 255.255.252.0;
  target-parameters TP2_v3_sha_none;
```

```
}
target-parameters TP2_v3_sha_none {
  parameters {
    message-processing-model v3;
    security-model usm;
    security-level none;
    security-name RU2_v3_sha_none;
  }
  notify-filter nfl;
}
notify N1_all_tl1_informs {
  type inform; # Replace "inform" with "trap" to convert informs to traps.
  tag tl1;
}
notify-filter nfl {
  oid .1 include;
}
view all {
  oid .1 include;
}
}
```

You can convert the SNMPv3 informs to traps by setting the value of the **type** statement at the **[edit snmp v3 notify N1_all_tl1_informs]** hierarchy level to **trap** as shown in the following example:

```
user@host# set snmp v3 notify N1_all_tl1_informs type trap
```

Related Documentation

SNMP Interaction with Juniper Networks Devices FAQs

This section presents frequently asked questions and answers related to how SNMP interacts with Juniper Networks devices.

How frequently should a device be polled? What is a good polling rate?

It is difficult to give an absolute number for the rate of SNMP polls per second since the rate depends on the following two factors:

- The number of variable bindings in a protocol data unit (PDU)
- The response time for an interface from the Packet Forwarding Engine

In a normal scenario where no delay is being introduced by the Packet Forwarding Engine and there is one variable per PDU (a Get request), the response time is 130+ responses per second. However, with multiple variables in an SNMP request PDU (30 to 40 for GetBulk requests), the number of responses per second is much less. Because the Packet Forwarding Engine load can vary for each system, there is greater variation in how frequently a device should be polled.

Frequent polling of a large number of counters, especially statistics, can impact the device. We recommend the following optimization on the SNMP managers:

- Use the row-by-row polling method, not the column-by-column method.
- Reduce the number of variable bindings per PDU.
- Increase timeout values in polling and discovery intervals.
- Reduce the incoming packet rate at the SNMP process (snmpd).

For better SNMP response on the device, the Junos OS does the following:

- Filters out duplicate SNMP requests.
- Excludes interfaces that are slow in response from SNMP queries.

One way to determine a rate limit is to note an increase in the **Currently Active** count from the **show snmp statistics extensive** command.

The following is a sample output of the **show snmp statistics extensive** command:

```
user@host> show snmp statistics extensive
SNMP statistics:
Input:
  Packets: 226656, Bad versions: 0, Bad community names: 0,
  Bad community uses: 0, ASN parse errors: 0,
  Too big: 0, No such names: 0, Bad values: 0,
  Read only: 0, General errors: 0,
  Total request varbinds: 1967606, Total set varbinds: 0,
  Get requests: 18478, Get nexts: 75794, Set requests: 0,
  Get responses: 0, Traps: 0,
  Silent drops: 0, Proxy drops: 0, Commit pending drops: 0,
  Throttle drops: 27084, Duplicate request drops: 0
V3 Input:
  Unknown security models: 0, Invalid messages: 0
  Unknown pdu handlers: 0, Unavailable contexts: 0
  Unknown contexts: 0, Unsupported security levels: 0
  Not in time windows: 0, Unknown user names: 0
  Unknown engine ids: 0, Wrong digests: 0, Decryption errors: 0
Output:
  Packets: 226537, Too big: 0, No such names: 0,
  Bad values: 0, General errors: 0,
  Get requests: 0, Get nexts: 0, Set requests: 0,
  Get responses: 226155, Traps: 382
SA Control Blocks:
  Total: 222984, Currently Active: 501, Max Active: 501,
  Not found: 0, Timed Out: 0, Max Latency: 25
SA Registration:
  Registers: 0, Deregisters: 0, Removes: 0
Trap Queue Stats:
  Current queued: 0, Total queued: 0, Discards: 0, Overflows: 0
Trap Throttle Stats:
  Current throttled: 0, Throttles needed: 0
Snmp Set Stats:
  Commit pending failures: 0, Config lock failures: 0
  Rpc failures: 0, Journal write failures: 0
  Mgd connect failures: 0, General commit failures: 0
```

Does SNMP open dynamic UDP ports? Why?

The SNMP process opens two additional ports (sockets): one for IPv4 and one for IPv6. This enables the SNMP process to send traps.

I am unable to perform a MIB walk on the ifIndex. Why is this?

Any variable bindings or values with an access level of **not-accessible** cannot be queried directly because they are part of other variable bindings in the SNMP MIB table. The ifIndex has an access level of **not-accessible**. Therefore, it cannot be accessed directly because it is part of the variable bindings. However, the ifIndex can be accessed indirectly through the variable bindings.

I see SNMP_IPC_READ_ERROR messages when the SNMP process restarts on my system and also during Routing Engine switchover. Is this acceptable?

Yes, it is acceptable to see **SNMP_IPC_READ_ERROR** messages when the SNMP process is restarted, the system reboots, or during a Routing Engine switchover. If all the processes come up successfully and the SNMP operations are working properly, then these messages can be ignored.

What is the source IP address used in the response PDUs for SNMP requests? Can this be configured?

The source IP address used in the response PDUs for SNMP requests is the IP address of the outgoing interface to reach the destination. The source IP address cannot be configured for responses. It can only be configured for traps.

**Related
Documentation**

SNMP Traps and Informs FAQs

This section presents frequently asked questions and answers related to SNMP traps and informs.

Does the Junos OS impose any rate limiting on SNMP trap generation?

The Junos OS implements a trap-queuing mechanism to limit the number of traps that are generated and sent.

If a 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, 2, 4, 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 traps in the queue are deleted.

Junos OS also has a throttle threshold mechanism to control the number of traps sent (default 500 traps) during a particular throttle interval (default 5 seconds). This helps ensure consistency in trap traffic, especially when a large number of traps are generated due to interface status changes.

The throttle interval begins when the first trap arrives at the throttle. All traps within the throttle threshold value are processed, and traps exceeding the threshold value are queued. The maximum size of all trap queues (the throttle queue and the destination

queue) is 40,000 traps. The maximum size of any one queue is 20,000 traps. When a trap is added to the throttle queue, or if the throttle queue has exceeded the maximum size, the trap is moved to the top of the destination queue. Further attempts to send the trap from the destination queue are stopped for a 30-second period, after which the destination queue restarts sending the traps.



NOTE: For the Juniper Networks EX Series Ethernet Switch, the maximum size of all trap queues (the throttle queue and the destination queue) is 1,000 traps. The maximum size for any one queue on the EX Series is 500 traps.

I did not see a trap when I had a syslog entry with a critical severity. Is this normal? Can it be changed?

Not every syslog entry with critical severity is a trap. However, you can convert any syslog entry to a trap using the **event-options** statement.

The following example shows how to configure a **jnxSyslogTrap** whenever an **rdp_ldp_nbrdown** syslog entry message error occurs.

```
user@host> show event-options
policy snmptrap {
  events rdp_ldp_nbrdown;
  then {
    raise-trap;
  }
}
```

Are SNMP traps compliant with the Alarm Reporting Function (X.733) on the Junos OS?

No, SNMP traps on the Junos OS are not X.733 compliant.

Can I set up filters for traps or informs?

Traps and informs can be filtered based on the trap category and the object identifier. You can specify categories of traps to receive per host by using the **categories** statement at the **[edit snmp trap-group trap-group]** hierarchy level. Use this option when you want to monitor only specific modules of the Junos OS.

The following example shows a sample configuration for receiving only **link**, **vrrp-events**, **services**, and **otn-alarms** traps:

```
[edit snmp]
trap-group jnpr {
  categories {
    link;
    vrrp-events;
    services;
    otn-alarms;
  }
  targets {
    192.168.69.179;
  }
}
```

```
}
```

The Junos OS also has a more advanced filter option (**notify-filter**) for filtering specific traps or a group of traps based on their object identifiers.

The SNMPv3 configuration also supports filtering of SNMPv1 and SNMPv2 traps and excluding Juniper Networks enterprise-specific configuration management traps, as shown in the following configuration example:

```
[edit snmp]
v3 {
  vacm {
    security-to-group {
      security-model v2c {
        security-name sn_v2c_trap {
          group gr_v2c_trap;
        }
      }
    }
  }
  access {
    group gr_v2c_trap {
      default-context-prefix {
        security-model v2c {
          security-level none {
            read-view all;
            notify-view all;
          }
        }
      }
    }
  }
}

target-address TA_v2c_trap {
  address 10.209.196.166;
  port 9001;
  tag-list tg1;
  target-parameters TP_v2c_trap;
}

target-parameters TP_v2c_trap {
  parameters {
    message-processing-model v2c;
    security-model v2c;
    security-level none;
    security-name sn_v2c_trap;
  }
  notify-filter nf1;
}

notify v2c_notify {
  type trap;
  tag tg1;
}

notify-filter nf1 {
  oid .1.3.6.1.4.1.2636.4.5 exclude;
  oid .1 include;
}

snmp-community index1 {
```

```

community-name "$9$tDLi01h7Nbw2axN"; ## SECRET-DATA
security-name sn_v2c_trap;
tag tgl;
}
view all {
oid .1 include;
}
}

```

Can I simulate traps on a device?

Yes, you can use the **request snmp spoof-trap *trap name*** command for simulating a trap to the NMS that normally receives your device's traps. You can also add required values using the **variable-bindings** parameter.

The following example shows how to simulate a trap to the local NMS using variable bindings:

```

user@host> request snmp spoof-trap linkDown variable-bindings "ifIndex[116]=116,
ifAdminStatus[116]=1 ,ifOperStatus[116]=2 , ifName[116]=ge-1/0/1"

```

How do I generate a warm start SNMPv1 trap?

When the SNMP process is restarted under normal conditions, a warm start trap is generated if the system up time is more than 5 minutes. If the system up time is less than 5 minutes, a cold start trap is generated.

The NMS sees only the MIB OIDs and numbers, but not the names of the SNMP traps. Why?

Before the NMS can recognize the SNMP trap details, such as the names of the traps, it must first compile and understand the MIBs and then parse the MIB OIDs.

In the Junos OS, how can I determine to which category a trap belongs?

For a list of common traps and their categories, 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* document.

Can I configure a trap to include the source IP address?

Yes, you can configure the **source-address**, **routing-instance**, or **logical-instance** name for the source IP address using the **trap-options** command:

```

user@host> show snmp trap-options
source-address 10.1.1.1;

```

Can I create a custom trap?

Yes, you can use the **jnxEventTrap** event script to create customized traps as needed.

In the following example, a Junos OS operations (op) script is triggered when a **UI_COMMIT_NOT_CONFIRMED** event is received. The Junos OS op script matches the complete message of the event and generates an SNMP trap.

Example: Junos OS Op Script

```

version 1.0;

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

param $event;
param $message;

match / {

    /*
     * trapm utility wants the following characters in the value to be escaped
     * '[', ']', ' ', '=', and ','
     */
    var $event-escaped = {
        call escape-string($text = $event, $vec = '[] =,');
    }

    var $message-escaped = {
        call escape-string($text = $message, $vec = '[] =,');
    }

    <op-script-results> {
    var $rpc = <request-snmp-spoof-trap> {
        <trap> "jnxEventTrap";
        <variable-bindings> "jnxEventTrapDescr[0]='Event-Trap' , "
        _ "jnxEventAvAttribute[1]='event' , "
        _ "jnxEventAvValue[1]='" _ $event-escaped _ "' , "
        _ "jnxEventAvAttribute[2]='message' , "
        _ "jnxEventAvValue[1]='" _ $message-escaped _ "'";
    }

    var $res = jcs:invoke($rpc);
    }

    template escape-string ($text, $vec) {

        if (jcs:empty($vec)) {
            expr $text;

        } else {
            var $index = 1;
            var $from = substring($vec, $index, 1);
            var $changed-value = {
                call replace-string($text, $from) {
                    with $to = {
                        expr "\\\";
                        expr $from;
                    }
                }
            }

            call escape-string($text = $changed-value, $vec = substring($vec, $index
+ 1));
        }
    }
}

```



```

template replace-string ($text, $from, $to) {

    if (contains($text, $from)) {
        var $before = substring-before($text, $from);
        var $after = substring-after($text, $from);
        var $prefix = $before _ $to;

        expr $before;
        expr $to;
        call replace-string($text = $after, $from, $to);

    } else {
        expr $text;
    }
}

```

After creating your customized trap, you must configure a policy on your device to tell the device what actions to take after it receives the trap.

Here is an example of a configured policy under the **[edit event-options]** hierarchy:

```

[edit event-options]
user@host> show
policy trap-on-event {
    events UI_COMMIT_NOT_CONFIRMED;
    attributes-match {
        UI_COMMIT_NOT_CONFIRMED.message matches complete;
    }
    then {
        event-script ev-syslog-trap.junos-op {
            arguments {
                event UI_COMMIT_NOT_CONFIRMED;
                message "${$.message}";
            }
        }
    }
}

```

Can I disable link up and link down traps on interfaces?

Yes, link up and link down traps can be disabled in the interface configuration. To disable the traps, use the **no-traps** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** and **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchies for physical and logical interfaces.

```
(traps | no-traps);
```

I see the link up traps on logical interfaces, but I do not see the link down traps. Is this normal behavior?

For Ethernet and ATM types of interfaces, Junos OS does not send link down traps for a logical interface if the physical interface is down to prevent flooding alarms for the same root cause. However, when the physical interface and logical interfaces come back up, traps are sent indicating link up. This is because the physical interface coming up does not necessarily mean the logical interfaces are also coming up.

For SONET types of interfaces with PPP encapsulation, Junos OS does send link down traps for a logical interface if the physical interface is down. When the physical interface and logical interfaces come back up, traps are sent for both the physical and logical interfaces indicating link up.

For SONET types of interfaces with HDLC encapsulation, Junos OS does not send link down traps for a logical interface if the physical interface is down. When the physical interface and logical interfaces come back up, traps are sent for both the physical and logical interfaces indicating link up.

For channelize interfaces with PPP encapsulation, Junos OS does send link down traps for a logical interface if the physical interface is down. When the physical interface and logical interfaces come back up, traps are sent for both the physical and logical interfaces indicating link up.

For channelize interfaces with HDLC encapsulation, Junos OS does not send link down traps for a logical interface if the physical interface is down. When the physical interface and logical interfaces come back up, traps are sent for both the physical and logical interfaces indicating link up.

Related Documentation

Junos OS Dual Routing Engine Configuration FAQs

This section presents frequently asked questions and answers related to the configuration of dual Routing Engines.

The SNMP configuration should be identical between the Routing Engines when configuring for continued communication. However, we recommend having separate Routing Engine IDs configured for each Routing Engine, when using SNMPv3.

In my system, the MIB object `snmpEngineBoots` is not in sync between two Routing Engines in a dual Routing Engine device. Is this normal behavior?

Yes. This is the normal behavior. Each Routing Engine runs its own SNMP process (`snmpd`) agent, allowing each Routing Engine to maintain its own engine boots.

Is there a way to identify that an address belongs to RE0, RE1, or the master Routing Engine management interface (`fxp0`) by looking at an SNMP walk?

No. When you do an SNMP walk on the device, it only displays the master Routing Engine management interface address.

What is the best way to tell if the current IP address belongs to `fxp0` or a Routing Engine, from a CLI session?

Routing Engines are mapped with the `fxp0` interface. This means that when you query RE0, the `ifTable` reports the `fxp0` interface address of RE0 only. Similarly, if you query RE1, the `ifTable` reports the `fxp0` interface address of RE1 only.

When there is a failover, the master hostname is changed since the hostname belongs to the Routing Engine. Is this correct?

Yes. You can configure the same hostname or different hostnames. Either would work.

If only the master IP address is configured (for example, 192.168.2.5), and the **sysDescr.0** object has the same string configured on both of the Routing Engines, then even after a switchover, the **sysDescr.0** object returns the same value. The following sample shows the results you get by using the **snmpget** command:

```
bng-junos-pool02: /c/svivek/PR_BRANCH/src> snmpget -c jnpr -v2c 192.168.2.5
sysDescr.0 system.sysDescr.0 = foo
```

SNMP Support for Routing Instances FAQs

This section presents frequently asked questions and answers related to how SNMP supports routing instances.

Can the SNMP manager access data for routing instances?

Yes, the 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.

Two different routing instance behaviors can occur, depending on where the clients originate:

- 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.

Routing instances are identified by either the context field in SNMPv3 requests or encoded in the community string in SNMPv1 or SNMPv2c 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 (including views, source address restrictions, and access privileges) is applied according to the actual community string (the set of data after the @ character—in this case **public**). However, if the community string **RI@public** is configured, the PDU is processed according to that community, and the embedded routing instance name is ignored.

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

Additionally, 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 SNMPv3 requests, the name **logical system/routing instance** should be identified directly in the context field.

Can I access a list of all routing instances on a device?

Yes, you can access a list of all the routing instances on a device using the `vacmContextName` object in the SNMP-VIEW-BASED-ACM MIB. In SNMP, each routing instance becomes a VACM context; this is why the routing instances appear in the `vacmContextName` object.

Can I access a default routing instance from a client in another logical router or routing instance?

No, the SNMP agent can only access data of the logical router to which it is connected.

Related Documentation

SNMP Counters FAQs

This section presents frequently asked questions and answers related to SNMP counters.

Which MIB should I use for interface counters?

Interface management over SNMP is based on two tables: the **ifTable** and its extension the **ifXTable**. Both are described in RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* and RFC 2233, *The Interfaces Group MIB using SMIv2*.

Interfaces can have several layers, depending on the media, and each sublayer is represented by a separate row in the table. The relationship between the higher layer and lower layers is described in the **ifStackTable**.

The **ifTable** defines 32-bit counters for inbound and outbound octets (`ifInOctets/ifOutOctets`), packets (`ifInUcastPkts/ifOutUcastPkts`, `ifInNUcastPkts/ifOutNUcastPkts`), errors, and discards.

The **ifXTable** provides similar 64-bit counters, also called high capacity (HC) counters, for inbound and outbound octets (`ifHCInOctets/ifHCOctets`) and inbound packets (`ifHCInUcastPkts`).

When should 64-bit counters be used?

It is always good to use 64-bit counters because they contain statistics for both low and high capacity components.

Are the SNMP counters `ifInOctets` and `ifOutOctets` the same as the command `reference show interfaces statistics in and out counters`?

Yes, these are the same, but only if SNMP is enabled when the router boots up. If you power on a Juniper Networks device and then enable SNMP, the SNMP counters start

from 0. SNMP counters do not automatically receive their statistics from the **show** command output. Similarly, using the **clear statistics** command does not clear the statistics that the SNMP counters collected, which can cause a discrepancy in the data that is seen by both processes.

Do the SNMP counters ifInOctets and ifOutOctets include the framing overhead for Point-to-Point Protocol (PPP) and High-Level Data Link Control (HDLC)?

Yes.

**Related
Documentation**

PART 3

Remote Monitoring (RMON) with SNMP

- [RMON Overview on page 217](#)
- [Configuring RMON Alarms and Events on page 221](#)
- [Monitoring RMON Alarms and Events on page 229](#)
- [Using RMON to Monitor Network Service Quality on page 235](#)

CHAPTER 11

RMON Overview

- [Understanding RMON Alarms on page 217](#)
- [Understanding RMON Events on page 219](#)

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 217](#)
- [jnxRmonAlarmTable on page 218](#)

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/junos13.3/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 *SNMP MIBs and Traps Reference*.

Related Documentation

- [Understanding RMON Events on page 219](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Using alarmTable to Monitor MIB Objects on page 229](#)

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 219](#)

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 217](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)
- [Using eventTable to Log Alarms on page 232](#)

Configuring RMON Alarms and Events

- [Understanding RMON Alarms and Events Configuration on page 221](#)
- [Minimum RMON Alarm and Event Entry Configuration on page 222](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)
- [Example: Configuring an RMON Alarm and Event Entry on page 227](#)

Understanding RMON Alarms and Events Configuration

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 217](#)
- [Understanding RMON Events on page 219](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)
- [Using alarmTable to Monitor MIB Objects on page 229](#)
- [Using eventTable to Log Alarms on page 232](#)
- [Minimum RMON Alarm and Event Entry Configuration on page 222](#)

Minimum RMON Alarm and Event Entry Configuration

To enable RMON on the router, you must configure an alarm entry and an event entry. To do this, include the following statements at the **[edit snmp rmon]** hierarchy level:

```
[edit snmp rmon]
alarm index {
  rising-event-index index;
  rising-threshold integer;
  sample-type type;
  variable oid-variable;
}
event index;
```

Related Documentation

- [Understanding RMON Alarms and Events Configuration on page 221](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)

Configuring an RMON Alarm Entry and Its Attributes

An alarm entry monitors the value of a MIB variable. You can configure how often the value is sampled, the type of sampling to perform, and what event to trigger if a threshold is crossed.

This section discusses the following topics:

- [Configuring the Alarm Entry on page 223](#)
- [Configuring the Description on page 223](#)
- [Configuring the Falling Event Index or Rising Event Index on page 223](#)
- [Configuring the Falling Threshold or Rising Threshold on page 224](#)
- [Configuring the Interval on page 224](#)
- [Configuring the Falling Threshold Interval on page 224](#)
- [Configuring the Request Type on page 225](#)
- [Configuring the Sample Type on page 225](#)
- [Configuring the Startup Alarm on page 225](#)

- [Configuring the System Log Tag on page 226](#)
- [Configuring the Variable on page 226](#)

Configuring the Alarm Entry

An alarm entry monitors the value of a MIB variable. The **rising-event-index**, **rising-threshold**, **sample-type**, and **variable** statements are mandatory. All other statements are optional.

To configure the alarm entry, include the **alarm** statement and specify an index at the **[edit snmp rmon]** hierarchy level:

```
[edit snmp rmon]
alarm index {
  description description;
  falling-event-index index;
  falling-threshold integer;
  falling-threshold-interval seconds;
  interval seconds;
  rising-event-index index;
  rising-threshold integer;
  sample-type (absolute-value | delta-value);
  startup-alarm (falling-alarm | rising alarm | rising-or-falling-alarm);
  variable oid-variable;
}
```

index is an integer that identifies an alarm or event entry.

Configuring the Description

The description is a text string that identifies the alarm entry.

To configure the description, include the **description** statement and a description of the alarm entry at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
description description;
```

Configuring the Falling Event Index or Rising Event Index

The falling event index identifies the event entry that is triggered when a falling threshold is crossed. The rising event index identifies the event entry that is triggered when a rising threshold is crossed.

To configure the falling event index or rising event index, include the **falling-event-index** or **rising-event-index** statement and specify an index at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
falling-event-index index;
rising-event-index index;
```

index can be from 0 through 65,535. The default for both the falling and rising event index is 0.

Configuring the Falling Threshold or Rising Threshold

The falling threshold is the lower threshold for the monitored variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup alarm is equal to **falling-alarm** or **rising-or-falling-alarm**. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising threshold. You must specify the falling threshold as an integer. Its default is 20 percent less than the rising threshold.

By default, the rising threshold is 0. The rising threshold is the upper threshold for the monitored variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated **startup-alarm** is equal to **rising-alarm** or **rising-or-falling-alarm**. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling threshold. You must specify the rising threshold as an integer.

To configure the falling threshold or rising threshold, include the **falling-threshold** or **rising-threshold** statement at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
falling-threshold integer;
rising-threshold integer;
```

integer can be a value from -2,147,483,647 through 2,147,483,647.

Configuring the Interval

The interval represents the period of time, in seconds, over which the monitored variable is sampled and compared with the rising and falling thresholds.

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

```
[edit snmp rmon alarm index]
interval seconds;
```

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

Configuring the Falling Threshold Interval

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



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

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

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

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

Configuring the Request Type

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

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

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

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

Configuring the Sample Type

The sample type identifies the method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is **absolute-value**, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is **delta-value**, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds.

To configure the sample type, include the **sample-type** statement and specify the type of sample at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
  sample-type (absolute-value | delta-value);
```

- **absolute-value**—Actual value of the selected variable is compared against the thresholds.
- **delta-value**—Difference between samples of the selected variable is compared against the thresholds.

Configuring the Startup Alarm

The startup alarm identifies the type of alarm that can be sent when this entry is first activated. You can specify it as **falling-alarm**, **rising-alarm**, or **rising-or-falling-alarm**.

To configure the startup alarm, include the **startup-alarm** statement and specify the type of alarm at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
  startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);
```

- **falling-alarm**—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.
- **rising-alarm**—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.
- **rising-or-falling-alarm**—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.

The default is **rising-or-falling-alarm**.

Configuring the System Log Tag

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

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

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

Configuring the Variable

The variable identifies the MIB object that is being monitored.

To configure the variable, include the **variable** statement and specify the object identifier or object name at the **[edit snmp rmon alarm index]** hierarchy level:

```
[edit snmp rmon alarm index]
  variable oid-variable;
```

oid-variable is a dotted decimal (for example, 1.3.6.1.2.1.2.1.2.1.10.1) or MIB object name (for example, ifInOctets.1).

Configuring an RMON 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 221](#)
- [Understanding RMON Alarms on page 217](#)
- [Understanding RMON Events on page 219](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Minimum RMON Alarm and Event Entry Configuration on page 222](#)
- [Example: Configuring an RMON Alarm and Event Entry on page 227](#)

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 221](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)

- [Configuring an RMON Event Entry and Its Attributes on page 226](#)

Monitoring RMON Alarms and Events

- [Using alarmTable to Monitor MIB Objects on page 229](#)
- [Using eventTable to Log Alarms on page 232](#)

Using alarmTable to Monitor MIB Objects

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

- [Creating an Alarm Entry on page 229](#)
- [Configuring the Alarm MIB Objects on page 229](#)
- [Activating a New Row in alarmTable on page 232](#)
- [Modifying an Active Row in alarmTable on page 232](#)
- [Deactivating a Row in alarmTable on page 232](#)

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 230](#)
- [alarmVariable on page 230](#)
- [alarmSampleType on page 230](#)
- [alarmValue on page 230](#)
- [alarmStartupAlarm on page 230](#)
- [alarmRisingThreshold on page 231](#)

- [alarmFallingThreshold](#) on page 231
- [alarmOwner](#) on page 231
- [alarmRisingEventIndex](#) on page 231
- [alarmFallingEventIndex](#) on page 231

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 217](#)
- [Understanding RMON Events on page 219](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)

Using eventTable to Log Alarms

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

- [Creating an Event Entry on page 232](#)
- [Configuring the MIB Objects on page 233](#)
- [Activating a New Row in eventTable on page 234](#)
- [Deactivating a Row in eventTable on page 234](#)

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 233
- [eventCommunity](#) on page 233
- [eventOwner](#) on page 233
- [eventDescription](#) on page 234

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 217](#)
- [Understanding RMON Events on page 219](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)

Using RMON to Monitor Network Service Quality

- [Understanding RMON for Monitoring Service Quality on page 235](#)
- [Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 239](#)
- [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*.

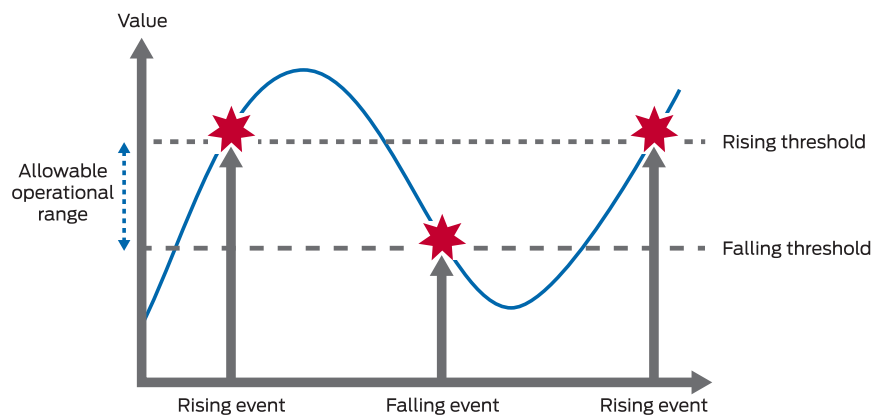
This topic includes the following sections:

- [Setting Thresholds on page 235](#)
- [RMON Command-Line Interface on page 236](#)
- [RMON Event Table on page 237](#)
- [RMON Alarm Table on page 237](#)
- [Troubleshooting RMON on page 238](#)

Setting Thresholds

By setting a rising and a falling threshold for a monitored variable, you can be alerted whenever the value of the variable falls outside of the allowable operational range. (See [Figure 3 on page 236](#).)

Figure 3: Setting Thresholds



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

To configure remote monitoring, specify the following pieces of information:

- The variable to be monitored (by its SNMP object identifier)
- The 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 18 on page 237](#).) 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 18: 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 19 on page 237](#).

Table 19: RMON Alarm Table

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

Table 19: 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 20 on page 238](#) to the RFC 2819 **alarmTable**.

Table 20: 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 239](#)

- [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 239](#)
- [Basic Key Performance Indicators on page 240](#)
- [Setting Baselines on page 240](#)

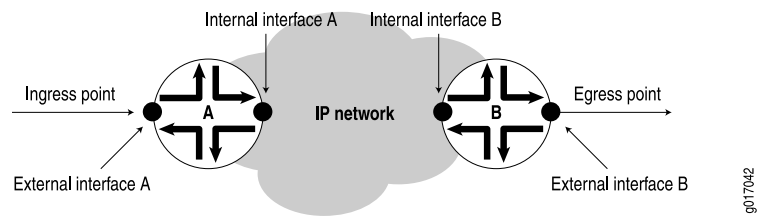
Measurement Points

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 4 on page 240](#).

- 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 4: Network Entry Points



NOTE: Figure 4 on page 240 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 235](#)
- [Defining and Measuring Network Availability on page 241](#)
- [Measuring Health on page 246](#)
- [Measuring Performance on page 253](#)

Defining and Measuring Network Availability

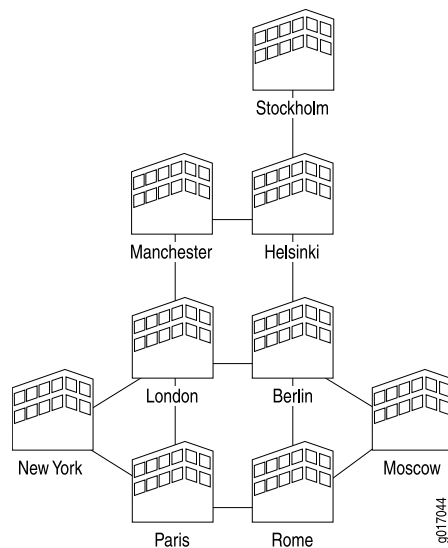
This topic includes the following sections:

- [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](#).

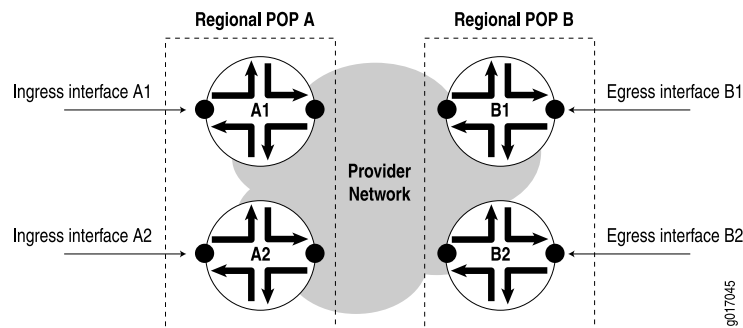
Figure 5: Regional Points of Presence



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:

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

To reduce management traffic on the service provider's network, instead of generating a full mesh of interface availability tests (for example, from each interface to every other interface), you can measure from each router's loopback address. This reduces the number of availability measurements required to a total of one for each router, or:

$[n \times (n-1)] / 2$ 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 21 on page 244](#).

Table 21: 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"> • <code>http-get</code> • <code>http-get-metadata</code> • <code>icmp-ping</code> • <code>icmp-ping-timestamp</code> • <code>tcp-ping</code> • <code>udp-ping</code>
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 21: 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 239](#)
- [Understanding RMON for Monitoring Service Quality on page 235](#)
- [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 22 on page 246](#).

Table 22: 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 22: 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 22: 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 22: 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 22: 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 22: 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 22: 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 23 on page 252](#).)

Table 23: 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](#)” on [page 62](#)” and “[Juniper Networks Enterprise-Specific SNMP Traps](#)” on [page 63](#)”.

- Related Documentation**
- [Understanding Measurement Points, Key Performance Indicators, and Baseline Values on page 239](#)
 - [Understanding RMON for Monitoring Service Quality on page 235](#)
 - [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 24 on page 253](#)).

Table 24: 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 24: 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 24: 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

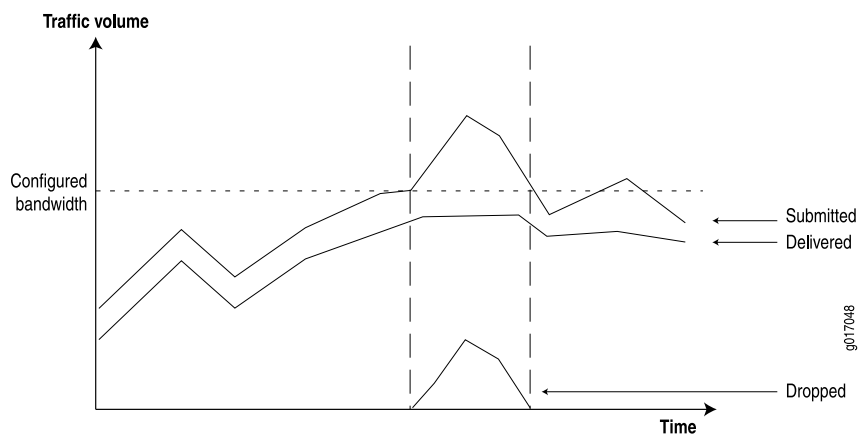
You can use class-of-service (CoS) mechanisms to regulate how certain classes of packets are handled within your network during times of peak congestion. Typically you must perform the following steps when implementing a CoS mechanism:

- 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 25 on page 257](#) shows additional filters used to match the other classes.

Table 25: 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 26 on page 257](#).

Table 26: 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 27 on page 258](#).)

Table 27: 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 28 on page 258](#).

Table 28: 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 29 on page 258](#).

Table 29: Dropped Traffic Counters

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

Table 29: 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 239](#)
- [Understanding RMON for Monitoring Service Quality on page 235](#)
- [Defining and Measuring Network Availability on page 241](#)
- [Measuring Health on page 246](#)

PART 4

Health Monitoring with SNMP

- [Configuring Health Monitoring on page 263](#)

Configuring Health Monitoring

- [Configuring Health Monitoring on Devices Running Junos OS on page 263](#)
- [Example: Configuring Health Monitoring on page 266](#)

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 264](#)
- [Minimum Health Monitoring Configuration on page 265](#)
- [Configuring the Falling Threshold or Rising Threshold on page 265](#)
- [Configuring the Interval on page 265](#)
- [Log Entries and Traps on page 266](#)

Monitored Objects

When you configure the health monitor, monitoring information for certain object instances is available, as shown in [Table 30 on page 264](#).

Table 30: Monitored Object Instances

Object	Description
<code>jnxHrStoragePercentUsed.1</code>	Monitors the following file system on the router or switch: /dev/ad0s1a: This is the root file system mounted on <code>/</code> .
<code>jnxHrStoragePercentUsed.2</code>	Monitors the following file system on the router or switch: /dev/ad0s1e: 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>sysAppElmtRunCPU</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>sysAppElmtRunMemory</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 221](#)
- [Configuring an RMON Alarm Entry and Its Attributes on page 222](#)
- [Configuring an RMON Event Entry and Its Attributes on page 226](#)
- [Example: Configuring Health Monitoring on page 266](#)
- [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 263](#)

PART 5

Gathering Statistics for Accounting Purposes Using Accounting Options, Source Class Usage and Destination Class Usage Options

- [Accounting Options, Source Class Usage and Destination Class Usage Options Overview on page 269](#)
- [Configuring Accounting Options, Source Class Usage and Destination Class Usage Options on page 273](#)

Accounting Options, Source Class Usage and Destination Class Usage Options Overview

- [Accounting Options Overview on page 269](#)
- [Understanding Source Class Usage and Destination Class Usage Options on page 270](#)

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 31 on page 269](#).

Table 31: 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 274](#)
- [Configuring Accounting-Data Log Files on page 278](#)
- [Configuring the Interface Profile on page 281](#)
- [Configuring the Filter Profile on page 283](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 273](#)

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 lookups 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 lookups 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 Core Routers and M320 Multiservice Edge 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 devices, 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.

On MX Series platforms with MPC/MIC interfaces, SCU and DCU are performed after output filters are evaluated. Packets dropped by output filters are not included in SCU or DCU statistics.

On MX Series platforms with non-MPC/MIC interfaces, SCU and DCU are performed before output filters are evaluated. Packets dropped by output filters are included in SCU and DCU statistics.

On Enhanced Scaling FPCs (T640-FPC1-ES, T640-FPC2-ES, T640-FPC3-ES, T640-FPC4-1P-ES, and T1600-FPC4-ES), the source class accounting is performed at ingress. Starting with Junos OS 13.3R2, the SCU accounting is performed at ingress on a T4000 Type 5 FPC. The implications of this are as follows:

- SCU accounting is performed when packets traverse from T4000 Type 5 FPC (ingress FPC) to Enhanced Scaling FPCs (egress FPC).
- SCU accounting is performed when packets traverse from Enhanced Scaling FPCs (ingress FPC) to T4000 Type 5 FPC (egress FPC).

Class-based filter match conditions are not supported on J Series Services Routers.

For more information about source class usage, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*, the *Junos OS Network Interfaces Library for Routing Devices*, and the *Junos OS, Release 14.1*.

**Related
Documentation**

- [Example: Grouping Source and Destination Prefixes into a Forwarding Class](#)
- [Configuring SCU or DCU on page 287](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 290](#)
- [Configuring Class Usage Profiles on page 291](#)
- [Configuring the MIB Profile on page 293](#)
- [Configuring the Routing Engine Profile on page 295](#)

Configuring Accounting Options, Source Class Usage and Destination Class Usage Options

- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 273
- Accounting Options Configuration on page 274
- Configuring Accounting-Data Log Files on page 278
- Configuring the Interface Profile on page 281
- Configuring the Filter Profile on page 283
- Example: Configuring a Filter Profile on page 285
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 286
- Configuring SCU or DCU on page 287
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 290
- Configuring Class Usage Profiles on page 291
- Configuring the MIB Profile on page 293
- Configuring the Routing Engine Profile on page 295

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.

```
[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 269](#)
 - [Accounting Options Configuration on page 274](#)

Accounting Options Configuration

This topic contains the following sections:

- [Accounting Options—Full Configuration on page 275](#)
- [Minimum Accounting Options Configuration on page 276](#)

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 (Accounting Options) 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.



NOTE: Do not configure MIB objects related to interface octets or packets for a MIB profile, because it can cause the SNMP walk or a CLI show command to time out.

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 (Accounting Options) 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 Library for Routing Devices*.

```

[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 269](#)
- [Understanding Device Management Functions in Junos OS on page 3](#)
- [Configuring Accounting-Data Log Files on page 278](#)
- [Configuring the Interface Profile on page 281](#)
- [Configuring the Filter Profile on page 283](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 273](#)

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 278](#)
- [Configuring the Maximum Size of the File on page 279](#)
- [Configuring the Maximum Number of Files on page 279](#)
- [Configuring the Start Time for File Transfer on page 279](#)
- [Configuring the Transfer Interval of the File on page 279](#)
- [Configuring Archive Sites on page 280](#)

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 Administration Library for Routing Devices*. 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 269](#)
- [Understanding Device Management Functions in Junos OS on page 3](#)
- [Accounting Options Configuration on page 274](#)
- [Configuring the Interface Profile on page 281](#)
- [Configuring the Filter Profile on page 283](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 273](#)

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;
}
```

By default, the Packet Forwarding Engine (PFE) periodically collects the statistics for all interfaces. To improve the performance, you can optionally disable the periodic refresh by including the **periodic-refresh disable** statement at the **[edit accounting-options]** hierarchy level.

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 an accounting profile at the **[edit firewall family family-type filter filter-name]** hierarchy level. For more information, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.

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

- [Configuring Fields on page 281](#)
- [Configuring the File Information on page 281](#)
- [Configuring the Interval on page 282](#)
- [Example: Configuring the Interface Profile on page 282](#)

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 269](#)
- [Understanding Device Management Functions in Junos OS on page 3](#)
- [Accounting Options Configuration on page 274](#)
- [Configuring Accounting-Data Log Files on page 278](#)
- [Configuring the Filter Profile on page 283](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 273](#)

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 Library for Routing Devices*.

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

- [Configuring the Counters on page 284](#)
- [Configuring the File Information on page 284](#)
- [Configuring the Interval on page 285](#)

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: The limit on the total number of characters per line in a log file equals 1023. If this limit is exceeded, the output written to the log file is incomplete. Ensure that you limit the number of counters or requested data so that this character limit is not exceeded.



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 269](#)
- [Understanding Device Management Functions in Junos OS on page 3](#)
- [Accounting Options Configuration on page 274](#)
- [Configuring Accounting-Data Log Files on page 278](#)
- [Configuring the Interface Profile on page 281](#)
- [Configuration Statements at the \[edit accounting-options\] Hierarchy Level on page 273](#)
- [Example: Configuring a Filter Profile on page 285](#)
- [Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 286](#)

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 283](#)
 - [Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 286](#)

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 283](#)
 - [Configuring the Interface Profile on page 281](#)

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 288](#)
- [Applying the Policy to the Forwarding Table on page 288](#)
- [Enabling Accounting on Inbound and Outbound Interfaces on page 288](#)

Creating Prefix Route Filters in a Policy Statement

To 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

To apply the policy to the forwarding table:

```
[edit]
routing-options {
  forwarding-table {
    export scu-1;
  }
}
```

Enabling Accounting on Inbound and Outbound Interfaces

To 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 as shown:

```
[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 *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices* and the *Junos OS Network Interfaces Library for Routing Devices*.

Related Documentation

- [Understanding Source Class Usage and Destination Class Usage Options on page 270](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 290](#)
- [Configuring Class Usage Profiles on page 291](#)
- [Configuring the MIB Profile on page 293](#)
- [Configuring the Routing Engine Profile on page 295](#)

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 290](#)
- [Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface on page 290](#)
- [Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface on page 291](#)

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 Library for Routing Devices*.

Related Documentation

- [Understanding Source Class Usage and Destination Class Usage Options on page 270](#)
- [Configuring SCU or DCU on page 287](#)
- [Configuring Class Usage Profiles on page 291](#)
- [Configuring the MIB Profile on page 293](#)
- [Configuring the Routing Engine Profile on page 295](#)

Configuring Class Usage Profiles

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

- [Configuring a Class Usage Profile on page 291](#)
- [Configuring the File Information on page 292](#)
- [Configuring the Interval on page 292](#)
- [Creating a Class Usage Profile to Collect Source Class Usage Statistics on page 292](#)
- [Creating a Class Usage Profile to Collect Destination Class Usage Statistics on page 293](#)

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-profile;  
    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 270](#)
- [Configuring SCU or DCU on page 287](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 290](#)
- [Configuring the Routing Engine Profile on page 295](#)

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 294](#)
- [Configuring the Interval on page 294](#)
- [Configuring the MIB Operation on page 294](#)
- [Configuring MIB Object Names on page 295](#)
- [Example: Configuring a MIB Profile on page 295](#)

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.



NOTE: Do not configure MIB objects related to interface octets or packets for a MIB profile, because it can cause the SNMP walk or a CLI show command to time out.

Example: Configuring a MIB Profile

Configure a MIB profile:

```
[edit accounting-options]
mib-profile mstatistics {
  file stats;
  interval 60;
  operation walk;
  objects-names {
    ipCidrRouteStatus;
  }
}
```

Related Documentation

- [Understanding Source Class Usage and Destination Class Usage Options on page 270](#)
- [Configuring SCU or DCU on page 287](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 290](#)
- [Configuring Class Usage Profiles on page 291](#)
- [Configuring the Routing Engine Profile on page 295](#)

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 296](#)
- [Configuring the File Information on page 296](#)
- [Configuring the Interval on page 296](#)
- [Example: Configuring a Routing Engine Profile on page 296](#)

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 270](#)
- [Configuring SCU or DCU on page 287](#)
- [Configuring SCU on a Virtual Loopback Tunnel Interface on page 290](#)
- [Configuring Class Usage Profiles on page 291](#)
- [Configuring the MIB Profile on page 293](#)

PART 6

Configuration Statements and Operational Commands

- Configuration Statements: Accounting Options, Source Class Usage and Destination Class Usage Options on page 301
- Configuration Statements: Health Monitoring on page 319
- Configuration Statements: Remote Monitoring (RMON) on page 323
- Configuration Statements: SNMP on page 335
- Configuration Statements: SNMPv3 on page 365
- Operational Commands on page 411

CHAPTER 18

Configuration Statements: Accounting Options, Source Class Usage and Destination Class Usage Options

- [accounting-options on page 302](#)
- [archive-sites on page 302](#)
- [class-usage-profile on page 303](#)
- [counters on page 304](#)
- [destination-classes on page 304](#)
- [fields \(for Interface Profiles\) on page 305](#)
- [fields \(for Routing Engine Profiles\) on page 306](#)
- [file \(Associating with a Profile\) on page 307](#)
- [file \(Configuring a Log File\) on page 308](#)
- [files on page 309](#)
- [filter-profile on page 310](#)
- [interface-profile on page 311](#)
- [interval \(Accounting Options\) on page 312](#)
- [mib-profile on page 313](#)
- [nonpersistent on page 314](#)
- [object-names on page 314](#)
- [operation on page 315](#)
- [routing-engine-profile on page 315](#)
- [size on page 316](#)
- [source-classes on page 316](#)
- [start-time \(Log File Transfer\) on page 317](#)
- [transfer-interval on page 317](#)

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 273• Accounting Options Configuration on page 274

archive-sites

Syntax	archive-sites { <i>site-name</i> ; }
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 <i>Getting Started Guide for Routing Devices</i> .
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 280

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	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
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 OS Routing Protocols Library for Routing Devices. For information about configuring source class usage, see Network Management Administration Guide for Routing Devices.</p>
Options	<p>profile-name—Name of the destination class 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 Class Usage Profiles on page 291

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 <code>/var/log</code> 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 284

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 291

fields (for Interface Profiles)

Syntax	<pre>fields { <i>field-name</i>; }</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	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 281

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 295

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 281• Configuring the Filter Profile on page 283• Configuring the MIB Profile on page 293• Configuring the Routing Engine Profile on page 295

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	<p><i>filename</i>—Name of the file in which to write accounting data.</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 Accounting-Data Log Files on page 278

files

Syntax	<code>files <i>number</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 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 278

filter-profile

Syntax	<pre>filter-profile <i>profile-name</i> { counters { counter-name; } 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 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 Firewall Filters Feature Guide for Routing Devices .
Options	<p>profile-name—Name of the filter 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 Filter Profile on page 283

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	<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 281

interval (Accounting Options)

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 281• Configuring the Filter Profile on page 283• Configuring the MIB Profile on page 293• Configuring the Routing Engine Profile on page 295

mib-profile

Syntax `mib-profile profile-name {
 file filename;
 interval minutes;
 object-names {
 mib-object-name;
 }
 operation operation-name;
 }`

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 `/var/log` directory.



NOTE: Do not configure MIB objects related to interface octets or packets for a MIB profile, because it can cause the SNMP walk or a CLI show command to time out.


Options *profile-name*—Name of the MIB 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 • [Configuring the MIB Profile on page 293](#)

nonpersistent

Syntax	nonpersistent;
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> 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.</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 278

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 293

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 293

routing-engine-profile

Syntax	<pre>routing-engine-profile profile-name { fields { field-name; } file filename; interval minutes; }</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 295

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	bytes —Maximum size of each log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0 , then profilelog.1 , and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. If you do not specify a size, the file is closed, archived, and renamed when the time specified for the transfer interval is exceeded. Syntax: <i>x</i> to specify bytes, <i>xk</i> to specify KB, <i>xm</i> to specify MB, <i>xg</i> to specify GB Range: 256 KB through 1 GB
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 279

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 291

start-time (Log File Transfer)

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: <code>YYYY-MM-DD.hh:mm</code>
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 279

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 279

Configuration Statements: Health Monitoring

- [falling-threshold](#) on page 319
- [health-monitor](#) on page 320
- [interval \(SNMP Health Monitor\)](#) on page 320
- [rising-threshold \(SNMP Health Monitor\)](#) on page 321

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 265 • rising-threshold (SNMP Health Monitor) on page 321

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 263

interval (SNMP Health Monitor)

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	<i>seconds</i> —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 265

rising-threshold (SNMP Health Monitor)

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 319• Configuring the Falling Threshold or Rising Threshold on page 265

CHAPTER 20

Configuration Statements: Remote Monitoring (RMON)

- [alarm \(SNMP RMON\) on page 324](#)
- [community \(SNMP RMON\) on page 325](#)
- [description \(SNMP RMON\) on page 325](#)
- [event \(SNMP\) on page 326](#)
- [falling-event-index on page 326](#)
- [falling-threshold on page 327](#)
- [falling-threshold-interval on page 328](#)
- [interval \(SNMP RMON\) on page 328](#)
- [request-type on page 329](#)
- [rising-event-index on page 330](#)
- [rising-threshold \(SNMP RMON\) on page 330](#)
- [rmon on page 331](#)
- [sample-type on page 331](#)
- [startup-alarm on page 332](#)
- [syslog-subtag on page 332](#)
- [type \(SNMP RMON\) on page 333](#)
- [variable on page 333](#)

alarm (SNMP RMON)

Syntax	<pre>alarm <i>index</i> { <i>description</i> <i>description</i>; <i>falling-event-index</i> <i>index</i>; <i>falling-threshold</i> <i>integer</i>; <i>falling-threshold-interval</i> <i>seconds</i>; <i>interval</i> <i>seconds</i>; <i>request-type</i> (get-next-request get-request walk-request); <i>rising-event-index</i> <i>index</i>; <i>rising-threshold</i> <i>integer</i>; <i>sample-type</i> (absolute-value delta-value); <i>startup-alarm</i> (falling-alarm rising-alarm rising-or-falling alarm); <i>syslog-subtag</i> <i>syslog-subtag</i>; <i>variable</i> <i>oid-variable</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. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure RMON alarm entries.
Options	<i>index</i> —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	<ul style="list-style-type: none">• Configuring an RMON Alarm Entry and Its Attributes on page 222• event (SNMP) on page 326• <i>Configuring RMON Alarms and Events</i>• <i>RMON MIB Event, Alarm, Log, and History Control Tables</i>• <i>Monitoring RMON MIB Tables</i>• <i>Understanding RMON</i>

community (SNMP RMON)

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 RMON Event Entry and Its Attributes on page 226

description (SNMP RMON)

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 223 • Configuring an RMON Event Entry and Its Attributes on page 226

event (SNMP)

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	index —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 RMON Event Entry and Its Attributes on page 226• alarm on page 324

falling-event-index

Syntax	<pre>falling-event-index <i>index</i>;</pre>
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	index —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 223• rising-event-index on page 330

falling-threshold

Syntax	<code>falling-threshold <i>integer</i>;</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	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	<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 224• rising-threshold (SNMP RMON) on page 330

falling-threshold-interval

Syntax	<code>falling-threshold-interval seconds;</code>
Hierarchy Level	<code>[edit snmp rmon alarm index]</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	seconds —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 224• interval (SNMP RMON) on page 328

interval (SNMP RMON)

Syntax	<code>interval seconds;</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	Interval between samples.
Options	seconds —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 224

request-type

Syntax	request-type (get-next-request get-request walk-request);
Hierarchy Level	[edit snmp rmon alarm index]
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 225• variable on page 333

rising-event-index

Syntax	<code>rising-event-index <i>index</i>;</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	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	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 223• falling-event-index on page 326

rising-threshold (SNMP RMON)

Syntax	<code>rising-threshold <i>integer</i>;</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	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	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 224• falling-threshold on page 327

rmon

Syntax	rmon { ... }
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 Remote Monitoring.
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 RMON Alarm Entry and Its Attributes on page 222

sample-type

Syntax	sample-type (absolute-value delta-value);
Hierarchy Level	[edit snmp rmon alarm index]
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	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 Sample Type on page 225

startup-alarm

Syntax	startup-alarm (falling-alarm rising-alarm rising-or-falling-alarm);
Hierarchy Level	[edit snmp rmon alarm index]
Release Information	Statement introduced before Junos 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 225

syslog-subtag

Syntax	syslog-subtag <i>syslog-subtag</i> ;
Hierarchy Level	[edit snmp rmon alarm index]
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 226

type (SNMP RMON)

Syntax	<code>type type;</code>
Hierarchy Level	[edit snmp rmon event index]
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	<p>type—Type of notification:</p> <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. <p>Default: log-and-trap</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 an RMON Event Entry and Its Attributes on page 226

variable

Syntax	<code>variable oid-variable;</code>
Hierarchy Level	[edit snmp rmon alarm index]
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	<p>oid-variable—OID of the MIB variable that is being monitored. The OID can be a dotted decimal (for example, 1.3.6.1.2.1.2.1.10.1). Alternatively, use the MIB object name (for example, ifInOctets.1).</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 Variable on page 226

CHAPTER 21

Configuration Statements: SNMP

- [access-list on page 336](#)
- [agent-address on page 337](#)
- [alarm-id on page 338](#)
- [alarm-list-name on page 339](#)
- [alarm-management on page 340](#)
- [alarm-state on page 341](#)
- [authorization on page 342](#)
- [categories on page 342](#)
- [client-list on page 343](#)
- [client-list-name on page 343](#)
- [clients on page 344](#)
- [commit-delay on page 344](#)
- [community \(SNMP\) on page 345](#)
- [contact \(SNMP\) on page 346](#)
- [description \(SNMP\) on page 347](#)
- [destination-port on page 348](#)
- [enterprise-oid on page 348](#)
- [filter-duplicates on page 349](#)
- [filter-interfaces on page 350](#)
- [interface \(SNMP\) on page 351](#)
- [location \(SNMP\) on page 351](#)
- [logical-system \(SNMP\) on page 352](#)
- [logical-system-trap-filter on page 353](#)
- [name on page 353](#)
- [nonvolatile on page 354](#)
- [oid on page 354](#)
- [proxy \(snmp\) on page 355](#)
- [routing-instance \(SNMP\) on page 356](#)

- [routing-instance-access on page 357](#)
- [snmp on page 357](#)
- [source-address \(SNMP\) on page 358](#)
- [targets on page 358](#)
- [traceoptions \(SNMP\) on page 359](#)
- [trap-group on page 361](#)
- [trap-options on page 362](#)
- [version \(SNMP\) on page 363](#)
- [view \(Associating a MIB View with a Community\) on page 363](#)
- [view \(Configuring a MIB View\) on page 364](#)

access-list

Syntax	<pre>[edit snmp] routing-instance-access { access-list { <i>routing-instance</i>; <i>routing-instance</i> restrict; } }</pre>
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 357

agent-address

Syntax	agent-address outgoing-interface;
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 91

alarm-id

Syntax `alarm-id id {
 alarm-state state {
 description alarm-description;
 notification-id notification-id-of-alarm;
 resource-prefix alarm-resource-prefix;
 varbind-index varbind-index-in-alarm-varbind-list;
 varbind-subtree alarm-varbind-subtree;
 varbind-value alarm-varbind-value;
 }
 }`

Hierarchy Level [edit snmp alarm-management alarm-list-name]

Release Information Statement introduced in Junos OS Release 14.1.

Description Specify the identifier of the alarm that you need to configure.

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

- [alarm-list-name on page 339](#)
- [alarm-management on page 340](#)
- [alarm-state on page 341](#)
- *jnxAlarmMib*

alarm-list-name

Syntax alarm-list-name *list-name* {
 alarm-id *id* {
 alarm-state *state* {
 description *alarm-description*;
 notification-id *notification-id-of-alarm*;
 resource-prefix *alarm-resource-prefix*;
 varbind-index *varbind-index-in-alarm-varbind-list*;
 varbind-subtree *alarm-varbind-subtree*;
 varbind-value *alarm-varbind-value*;
 }
 }
 }

Hierarchy Level [edit snmp alarm-management]

Release Information Statement introduced in Junos OS Release 14.1.

Description Specify the name of the alarm list that you need to configure.

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

- [alarm-id on page 338](#)
- [alarm-management on page 340](#)
- [alarm-state on page 341](#)
- *jnxAlarmMib*

alarm-management

```
Syntax  alarm-management {
        alarm-list-name list-name {
            alarm-id id {
                alarm-state state {
                    description alarm-description;
                    notification-id notification-id-of-alarm;
                    resource-prefix alarm-resource-prefix;
                    varbind-index varbind-index-in-alarm-varbind-list;
                    varbind-subtree alarm-varbind-subtree;
                    varbind-value alarm-varbind-value;
                }
            }
        }
    }
```

Hierarchy Level [edit snmp]

Release Information Statement introduced in Junos OS Release 14.1.

Description Configure the alarm management system to monitor and report active alarms as well as the history of alarms through the SNMP MIB tables supported by the *Alarm MIB*.



NOTE: You cannot configure alarms without notifications. It is mandatory to include the notification identifier in the configuration.

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

- [alarm-id on page 338](#)
- [alarm-list-name on page 339](#)
- [alarm-state on page 341](#)
- *jnxAlarmMib*

alarm-state

Syntax	<pre>alarm-state state { description <i>alarm-description</i>; notification-id <i>notification-id-of-alarm</i>; resource-prefix <i>alarm-resource-prefix</i>; varbind-index <i>varbind-index-in-alarm-varbind-list</i>; varbind-subtree <i>alarm-varbind-subtree</i>; varbind-value <i>alarm-varbind-value</i>; }</pre>
Hierarchy Level	[edit snmp alarm-management alarm-list-name]
Release Information	Statement introduced in Junos OS Release 14.1.
Description	Specify the state of the alarm and the other parameters that you need to monitor.
Options	<p>description <i>alarm-description</i>—Include a brief description of the alarm.</p> <p>notification-id <i>notification-id-of-alarm</i>—Specify the identifier of the notification associated with the alarm.</p> <p>resource-prefix <i>alarm-resource-prefix</i>—Specify the resource prefix of the alarm.</p> <p>varbind-index <i>varbind-index-in-alarm-varbind-list</i>—Specify the varbind index in the alarm varbind list. Range: 0 through 4294967295</p> <p>varbind-subtree <i>alarm-varbind-subtree</i>—Specify the subtree of the varbind.</p> <p>varbind-value <i>alarm-varbind-value</i>—Specify the varbind value of the alarm. Range: 0 through 2147483647</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"> • alarm-id on page 338 • alarm-list-name on page 339 • alarm-management on page 340 • <i>jnxAlarmMib</i>

authorization

Syntax	<code>authorization <i>authorization</i>;</code>
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. 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	<i>authorization</i> —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: <code>read-only</code>
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 Communities on page 80

categories

Syntax	<code>categories { <i>category</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	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	<i>category</i> —Name of a trap type: authentication , chassis , configuration , link , remote-operations , rmon-alarm , routing , services , sonet-alarms , startup , or vrrp-events .
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 Trap Groups on page 92

client-list

Syntax	<code>client-list <i>client-list-name</i> { <i>ip-addresses</i>; }</code>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 8.5. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for QFX Series switches.
Description	Define a list of SNMP clients.
Options	<i>client-list-name</i> —Name of the client list. <i>ip-addresses</i> —IP addresses of the SNMP clients to be added to the client list,
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 83

client-list-name

Syntax	<code>client-list-name <i>client-list-name</i>;</code>
Hierarchy Level	[edit snmp community <i>community-name</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 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 83

clients

Syntax	<pre>clients { address <restrict>; }</pre>
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.
Description	Specify the IPv4 or IPv6 addresses of the SNMP client hosts that are authorized to use this community.
Default	If you omit the clients statement, all SNMP clients using this community string are authorized to access the router.
Options	address —Address of an SNMP client that is authorized to access this router. You must specify an address, not a hostname. To specify more than one client, include multiple address options. restrict —(Optional) Do not allow the specified SNMP client to access the router.
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 Communities on page 80

commit-delay

Syntax	<pre>commit-delay <i>seconds</i>;</pre>
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	seconds —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 79

community (SNMP)

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	<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 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>community-name—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 SNMP Communities on page 80

contact (SNMP)

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 75

description (SNMP)

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.
Default	By default, the sysDescription object includes the following information: Juniper Networks, Inc. <i>platform</i> , <i>build</i> , Build date: <i>date</i> UTC Copyright (c) <i>date-range</i> Juniper Networks, Inc. For example: sysDescr.0 = Juniper Networks, Inc. m7i internet router, kernel JUNOS 13.2-20130530_ib_13_3_psd.1, Build date: 2013-05-30 22:48:07 UTC Copyright (c) 1996-2013 Juniper Networks, Inc.
Options	<i>description</i> —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 76

destination-port

Syntax	destination-port <i>port-number</i> ;
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 92


enterprise-oid

Syntax	enterprise-oid;
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 88

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 79

filter-interfaces

Syntax	<pre>filter-interfaces { all-internal-interfaces; interfaces <i>interface-names</i>{ interface 1; interface 2; } }</pre>
Hierarchy Level	[edit snmp]
Release Information	<p>Statement introduced in Junos OS Release 9.4.</p> <p>Statement introduced in Junos OS Release 9.4 for EX Series Switches.</p>
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 internal interfaces that include interfaces belonging to internal routing tables, such as <code>_juniper_private1_</code>, <code>_juniper_private2_</code>, <code>_juniper_private3_</code>, and <code>_juniper_private4_</code>. In addition, this option filters out information about the physical interface corresponding to a logical interface with a unit number from the internal routing table.</p> <p>interfaces—Specifies the interfaces to filter out from the output of SNMP Get and GetNext requests.</p>
	<div>  <p>NOTE: Starting with Release 12.1, Junos OS provides an except option (! operator) that enables you to filter out all interfaces except those interfaces that match the regular expressions prefixed with the ! mark.</p> </div>
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"> Filtering Interface Information Out of SNMP Get and GetNext Output on page 96


interface (SNMP)

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 95

location (SNMP)

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 76

logical-system (SNMP)

Syntax	<pre>logical-system <i>logical-system-name</i> { <i>routing-instance routing-instance-name</i>; }</pre>
Hierarchy Level	<pre>[edit snmp <i>community community-name</i>], [edit snmp <i>trap-group</i>], [edit snmp <i>trap-options</i>] [edit snmp <i>v3target-address target-address-name</i>]</pre>
Release Information	<p>Statement introduced in Junos OS Release 9.3</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>
<div>  <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> </div>	
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>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"> • Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157 • Configuring the Trap Target Address on page 124

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"> • SNMP Traps Supported for Routing Instances on page 155

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 78

nonvolatile

Syntax	<code>nonvolatile { commit-delay <i>seconds</i>; }</code>
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 79• commit-delay on page 344

oid

Syntax	<code>oid <i>object-identifier</i> (exclude include);</code>
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	exclude —Exclude the subtree of MIB objects represented by the specified OID. include —Include the subtree of MIB objects represented by the specified OID. 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.
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 97

proxy (snmp)

Syntax	<pre> proxy <i>proxy-name</i>{ device-name <i>device-name</i>; logical-system <i>logical-system</i> { routing-instance <i>routing-instance</i>; } routing-instance <i>routing-instance</i>; (version-v1 version-v2c) { snmp-community <i>community-name</i>; no-default-comm-to-v3-config; } version-v3 { security-name <i>security-name</i>; context <i>context-name</i>; } } </pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	Configure a device to act as a proxy SNMP agent, and specify a name for the proxy.
Options	<p>context <i>context-name</i>—Specify the SNMPv3 context name as configured on the device specified at edit snmp proxy <i>proxy-name</i> device-name <i>device-name</i>. For more information about this statement, see context.</p> <p>device-name <i>device-name</i>—Specify the name of the device to be managed through the proxy SNMP agent.</p> <p>no-default-comm-to-v3-config—(Optional) Specify whether you have to manually configure the statements at the [edit snmp v3 snmp-community <i>community-name</i>] and [edit snmp v3 vacm] hierarchy levels. If this statement is not included in the configuration, the [edit snmp v3 snmp-community <i>community-name</i>] and [edit snmp v3 vacm] hierarchy level configurations are automatically initialized.</p> <p><i>proxy-name</i>—Specify the name of the proxy.</p> <p>security-name <i>security-name</i>—Specify the SNMPv3 security name as configured on the device specified at edit snmp proxy <i>proxy-name</i> device-name <i>device-name</i>. For more information about this statement, see security-name.</p> <p>snmp-community <i>community-name</i>—Specify the name of the SNMP community. The community name you configure should match the snmp-community configuration on the device specified at edit snmp proxy <i>proxy-name</i> device-name <i>device-name</i>. For more information about this statement, see snmp-community.</p> <p>(version-v1 version-v2c)—Specify the SNMP version, and add the relevant configuration.</p>

version-v3—Add the SNMPv3 configuration.

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 a Proxy SNMP Agent on page 85

routing-instance (SNMP)

Syntax	routing-instance <i>routing-instance-name</i> ;
Hierarchy Level	[edit snmp community <i>community-name</i>], [edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i>], [edit snmp trap-group <i>group</i>]
Release Information	Statement introduced in Junos OS Release 8.3. Added to the [edit snmp community <i>community-name</i>] hierarchy level in Junos OS Release 8.4. Added to the [edit snmp community <i>community-name</i> logical-system <i>logical-system-name</i>] 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 [edit snmp community <i>community-name</i>] hierarchy level and specify the routing-instance statement under the [edit snmp community <i>community-name</i> logical-system <i>logical system-name</i>] 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 92• Configuring the Source Address for SNMP Traps on page 89• Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 157

routing-instance-access

Syntax	[edit snmp] routing-instance-access { access-list { <i>routing-instance</i> ; <i>routing-instance</i> restrict; } }
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 157

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. SNMP modules cannot have the slash (/) character or the @ character in the 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 SNMP on a Device Running Junos OS on page 73

source-address (SNMP)

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 89

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 92

traceoptions (SNMP)

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>; memory-trace; no-remote-trace; no-default-memory-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> <p>memory-trace and no-default-memory-trace options introduced in Junos OS Release 13.3.</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 • ilmid • 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>

- **all**—Log all SNMP 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.
- **protocol-timeouts**—Log SNMP response timeouts.
- **routing-socket**—Log routing socket calls.
- **subagent**—Log subagent restarts.
- **timer**—Log internally generated events.
- **varbind-error**—Log variable binding errors.

memory-trace—Enable tracing in memory. If **file** and **memory-trace** are configured, tracing will occur in file and memory.



NOTE: By default, memory tracing is enabled with default trace flags even after **snmp traceoptions** config statement is disabled.

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

no-default-memory-trace—(Optional) Disable default memory-tracing.

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 • [Tracing SNMP Activity on a Device Running Junos OS on page 181](#)

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 92

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 88


version (SNMP)

Syntax	<code>version (all v1 v2);</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	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 92

view (Associating a MIB View with a Community)

Syntax	<code>view <i>view-name</i>;</code>
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	Associate a view with a community. A view represents a group of MIB objects.
Options	<i>view-name</i> —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 SNMP Communities on page 80

view (Configuring a MIB View)

Syntax	<pre>view <i>view-name</i> { <i>oid object-identifier</i> (include exclude); }</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	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 <i>community-name</i>] hierarchy level.
	<div> NOTE: To remove an OID completely, use the <code>delete view all oid oid-number</code> command but omit the <code>include</code> parameter.</div>
Options	<p><i>view-name</i>—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 97• Associating MIB Views with an SNMP User Group on page 116• community on page 345

CHAPTER 22

Configuration Statements: SNMPv3

- [address \(SNMP\) on page 366](#)
- [address-mask on page 367](#)
- [authentication-md5 on page 367](#)
- [authentication-none on page 368](#)
- [authentication-password on page 369](#)
- [authentication-sha on page 370](#)
- [community-name \(SNMP\) on page 371](#)
- [context \(SNMPv3\) on page 372](#)
- [engine-id \(SNMP\) on page 373](#)
- [group \(Configuring Access Privileges\) on page 374](#)
- [group \(Associating a Security Name\) on page 375](#)
- [retry-count \(SNMPv3\) on page 375](#)
- [timeout \(SNMP\) on page 376](#)
- [local-engine on page 377](#)
- [message-processing-model on page 378](#)
- [notify on page 379](#)
- [notify-filter \(Applying to the Management Target\) on page 380](#)
- [notify-filter \(Configuring the Profile Name\) on page 380](#)
- [notify-view on page 381](#)
- [oid \(SNMP\) on page 381](#)
- [parameters on page 382](#)
- [port \(SNMP\) on page 382](#)
- [privacy-3des on page 383](#)
- [privacy-aes128 on page 384](#)
- [privacy-des on page 385](#)
- [privacy-none on page 385](#)
- [privacy-password on page 386](#)
- [read-view on page 387](#)

- [remote-engine](#) on page 388
- [routing-instance \(SNMPv3\)](#) on page 389
- [security-level \(Defining Access Privileges\)](#) on page 390
- [security-level \(Generating SNMP Notifications\)](#) on page 391
- [security-model \(Access Privileges\)](#) on page 392
- [security-model \(Group\)](#) on page 393
- [security-model \(SNMP Notifications\)](#) on page 394
- [security-name \(Community String\)](#) on page 395
- [security-name \(Security Group\)](#) on page 396
- [security-name \(SNMP Notifications\)](#) on page 397
- [security-to-group](#) on page 398
- [snmp-community](#) on page 399
- [tag \(SNMPv3\)](#) on page 399
- [tag-list](#) on page 400
- [target-address](#) on page 401
- [target-parameters](#) on page 402
- [type \(SNMPv3\)](#) on page 403
- [user](#) on page 403
- [usm](#) on page 404
- [v3](#) on page 406
- [vacm](#) on page 408
- [write-view](#) on page 409

address (SNMP)

Syntax	<code>address <i>address</i>;</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.
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	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 on page 125

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. Statement introduced in Junos OS Release 11.1 on the QFX Series.
Description	Define and verify the source addresses for a group of target addresses for SNMP traps and informs.
Options	<i>address-mask</i> —Define 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 125

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 111

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 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 No Authentication on page 112

authentication-password

Syntax	<code>authentication-password <i>authentication-password</i>;</code>
Hierarchy Level	<code>[edit snmp v3 usm local-engine user <i>username</i> authentication-md5],</code> <code>[edit snmp v3 usm local-engine user <i>username</i> authentication-sha],</code> <code>[edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> authentication-md5],</code> <code>[edit snmp v3 usm remote-engine <i>engine-id</i> user <i>username</i> authentication-sha]</code>
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	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 lowercase letters, uppercase letters, numbers, and the following special characters: <code>.,/\<>;:'[]{}~!@#\$%^*_+=-`</code> <p>In addition, the following special characters are also supported, but you must enclose them within quotation marks ("") if you enter them on the CLI; if you use a Network Management System to enter the password, the quotation marks are not required: <code> & () ?</code></p> <p>Control characters—entered by simultaneously pressing the Ctrl key and additional keys—are not supported.</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"> • Configuring MD5 Authentication on page 111 • Configuring SHA Authentication on page 111

authentication-sha

Syntax	<code>authentication-sha { authentication-password <i>authentication-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 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 111


community-name (SNMP)

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. Statement introduced in Junos OS Release 11. for the QFX Series.
Description	Define an SNMP community to authorize SNMPv1 or SNMPv2c clients in an SNMPv3 system. When you configure a community in SNMPv3, you can also specify a security name. The access privileges associated with the security name determine which MIB objects are available and which operations (read, write, or notify) are 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 the name 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 138

context (SNMPv3)

Syntax	context <i>context-name</i> ;
Hierarchy Level	[edit snmp v3 snmp-community <i>community-index</i>]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Specify the SNMPv3 context for access control. A context identifies a collection of information accessible for an SNMP entity.
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 SNMPv3 Community on page 138

engine-id (SNMP)

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 the management port.</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 108

group (Configuring Access Privileges)

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

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.

(Not applicable to the QFX Series.) 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. To specify a routing instance that is part of a logical system, specify it as **logical system/routing instance**. For example, to specify routing instance ri1 in logical system ls1, include **context-prefix ls1/ri1**.

The remaining statements under this hierarchy are explained separately.

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 115](#)

group (Associating a Security Name)

Syntax	<code>group <i>group-name</i>;</code>
Hierarchy Level	[edit snmp v3 vacm security-to-group security-model (usm v1 v2c) <code>security-name security-name</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 a security name with a group composed of users with the same access privileges. The security name is used during authentication of SNMP messages, and is mapped to a username.
Options	<i>group-name</i> —Collection of SNMP security names that share the same SNMPv3 access privileges.
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 120

retry-count (SNMPv3)

Syntax	<code>retry-count <i>number</i>;</code>
Hierarchy Level	[edit snmp v3 <code>target-address target-address-name</code>]
Release Information	Statement introduced in Junos OS Release 7.4. Statement introduced in Junos OS Release 11.1 for the QFX Series.
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 131 • timeout on page 376

timeout (SNMP)

Syntax	timeout <i>seconds</i> ;
Hierarchy Level	[edit snmp v3 target-address <i>target-address-name</i>]
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	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 131• retry-count (SNMPv3) on page 375

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 109](#)

message-processing-model

Syntax	message-processing-model (v1 v2c v3);
Hierarchy Level	[edit snmp v3 target-parameters <i>target-parameter-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 message processing model to be used when generating SNMP notifications.
Options	v1—SNMPv1 message process model. v2c—SNMPv2c message process model. v3—SNMPv3 message process model.
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 Message Processing Model on page 129

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 136 • Configuring the SNMPv3 Trap Notification on page 122

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. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Configure the notify filter applied to a specific set of SNMPv3 target parameters. Target parameters are the message processing and security parameters for notifications sent to a target SNMP manager.
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 129

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 124• oid (SNMP) on page 381

notify-view

Syntax	<code>notify-view <i>view-name</i>;</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.
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 97 • Configuring the Notify View on page 117

oid (SNMP)

Syntax	<code>oid <i>oid</i> (include exclude);</code>
Hierarchy Level	[edit snmp v3 notify-filter <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 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 124

parameters

Syntax	<pre>parameters { message-processing-model (v1 v2c v3); security-level (none authentication privacy); security-model (usm v1 v2c); security-name security-name; }</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 128

port (SNMP)

Syntax	<pre>port port-number;</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 126

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 SNMPv3 Encryption Type on page 112

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 SNMPv3 Encryption Type on page 112

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 SNMPv3 Encryption Type on page 112

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 SNMPv3 Encryption Type on page 112

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 SNMPv3 Encryption Type on page 112

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 117 • Configuring MIB Views on page 97

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	<p>Statement introduced 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	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	<p><i>engine-id</i>—Specify engine identifier in hexadecimal format. Used to compute the security digest for authenticating and encrypting packets sent to a user on the remote host.</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 Remote Engine and Remote User on page 132

routing-instance (SNMPv3)

Syntax	<code>routing-instance <i>routing-instance-name</i>;</code>
Hierarchy Level	[edit snmp v3 target-address <i>target-address-name</i>]
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	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 124

security-level (Defining Access Privileges)

Syntax	<pre>security-level (authentication none privacy) { notify-view view-name; read-view view-name; write-view view-name; }</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 116

security-level (Generating SNMP Notifications)

Syntax	security-level (authentication none privacy);
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 level to use when generating SNMP notifications.
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 130

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 116


security-model (Group)

Syntax	<pre>security-model (usm v1 v2c) { security-name security-name { group group-name; } }</pre>
Hierarchy Level	[edit snmp v3 vacm security-to-group]
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	Define a security model for an SNMPv3 group and associate the security name of a user with the group. All users in the group have the same access privileges.
Options	<p>usm—SNMPv3 security model.</p> <p>v1—SNMPv1 security model.</p> <p>v2c—SNMPv2c security model.</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 Security Model on page 119

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 130


security-name (Community String)

Syntax	<code>security-name <i>security-name</i>;</code>
Hierarchy Level	<code>[edit snmp v3 <i>snmp-community</i> <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. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Associate a community string with the security name of a user. The community string, which is used for SNMPv1 and SNMPv2c clients in an SNMPv3 system, is configured at the <code>[edit snmp v3 snmp-community <i>community-index</i>]</code> hierarchy level.
Options	<i>security-name</i> —Name that is used for messaging security and user access control.
<div>  <p>NOTE: The security name must match the configured security name at the <code>[edit snmp v3 target-parameters <i>target-parameters-name</i> parameters]</code> hierarchy level when you configure traps or informs.</p> </div>	
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 Names on page 140

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. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Associate the security name of a user (for SNMPv3 clients) or a community string (for SNMPv1 and SNMPv2c clients) with a configured security group.
Options	security-name —SNMPv3 secure username configured at the [edit snmp v3 usm local-engine user <i>username</i>] hierarchy level that is used for messaging security. For SNMPv1 and SNMPv2c, the security name is the community string configured at the [edit snmp v3 snmp-community community-index] 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 120• <i>Assigning a Security Name to a Group</i>

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>  <p>NOTE: The access privileges for the group associated with this security name must allow this notification to be sent.</p> <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> </div>	
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 Name on page 130

security-to-group

Syntax	<pre>security-to-group { security-model (usm v1 v2c) { group group-name; security-name security-name; } }</pre>
Hierarchy Level	[edit snmp v3 vacm]
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 group to which a specific SNMPv3 security name belongs. The security name is used for messaging 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">• Assigning Security Model and Security Name to a Group on page 119

snmp-community

Syntax	snmp-community <i>community-index</i> { community-name <i>community-name</i> ; security-name <i>security-name</i> ; tag <i>tag-name</i> ; }
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 the SNMP community which authorizes SNMPv1 or SNMPv2c clients in an SNMPv3 system.
Options	<i>community-index</i> —(Optional) String that identifies an SNMP community. 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 SNMPv3 Community on page 138

tag (SNMPv3)

Syntax	tag <i>tag-name</i> ;
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 140 • Configuring the SNMPv3 Trap Notification on page 122

tag-list

Syntax	<code>tag-list <i>tag-list</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. 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 126

target-address

Syntax	<pre>target-address <i>target-address-name</i> { address <i>address</i>; address-mask <i>address-mask</i>; logical-system (SNMP) <i>logical-system</i>; port (SNMP) <i>port-number</i>; retry-count (SNMPv3) <i>number</i>; routing-instance (SNMPv3) <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 124

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 128](#)
- [Applying Target Parameters on page 127](#)

type (SNMPv3)

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	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 131• Configuring the SNMPv3 Trap Notification on page 122

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	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">• Creating SNMPv3 Users on page 109

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 • [Creating SNMPv3 Users on page 109](#)
Documentation • [Configuring the Remote Engine and Remote User on page 132](#)

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 (SNMP) 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 104

vacm

Syntax	<pre> 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; } } } </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> <p>Statement introduced in Junos OS Release 11.1 for the QFX Series.</p>
Description	<p>Configure view-based access control model (VACM) information, including access privileges such as security model and security level for a group of users.</p> <p>The remaining statements are explained separately.</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"> • Defining Access Privileges for an SNMP Group on page 114

write-view

Syntax	<code>write-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 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	<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 MIB Views on page 97 • Configuring the Write View on page 118

CHAPTER 23

Operational Commands

- `clear ilmi statistics`
- `clear snmp history`
- `clear snmp statistics`
- `request snmp spoof-trap`
- `show interfaces snmp-index`
- `show ilmi statistics`
- `show snmp health-monitor`
- `show snmp inform-statistics`
- `show snmp mib`
- `show snmp rmon`
- `show snmp statistics`
- `show snmp v3`

clear ilmi statistics

Syntax	clear ilmi statistics
Release Information	Command introduced before Junos OS Release 7.4.
Description	Set Integrated Local Management Interface (ILMI) statistics to zero.
Options	This command has no options.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show ilmi statistics on page 423
List of Sample Output	clear ilmi statistics on page 412
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear ilmi statistics

```
user@host> clear ilmi statistics
```

clear snmp history

Syntax	clear snmp history (<i>index</i> all)
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Delete the history record of Simple Network Management Protocol (SNMP) samples of Ethernet statistics collected.
Options	all —Clear all the entries in the history index. index —Clear the contents of the specified entry in the history index.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• clear snmp statistics on page 414

clear snmp statistics

Syntax	clear snmp statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Clear Simple Network Management Protocol (SNMP) statistics.
Options	This command has no options.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show snmp statistics on page 442
List of Sample Output	clear snmp statistics on page 414
Output Fields	See show snmp statistics for an explanation of output fields.

Sample Output

clear snmp statistics

In the following example, SNMP statistics are displayed before and after the **clear snmp statistics** command is issued:

```
user@host> show snmp statistics
SNMP statistics:
  Input:
    Packets: 8, Bad versions: 0, Bad community names: 0,
    Bad community uses: 0, ASN parse errors: 0,
    Too bigs: 0, No such names: 0, Bad values: 0,
    Read onlys: 0, General errors: 0,
    Total request varbinds: 8, Total set varbinds: 0,
    Get requests: 0, Get nexts: 8, Set requests: 0,
    Get responses: 0, Traps: 0,
    Silent drops: 0, Proxy drops 0
  Output:
    Packets: 2298, Too bigs: 0, No such names: 0,
    Bad values: 0, General errors: 0,
    Get requests: 0, Get nexts: 0, Set requests: 0,
    Get responses: 8, Traps: 2290
```

```
user@host> clear snmp statistics
```

```
user@host> show snmp statistics
SNMP statistics:
  Input:
    Packets: 0, Bad versions: 0, Bad community names: 0,
    Bad community uses: 0, ASN parse errors: 0,
    Too bigs: 0, No such names: 0, Bad values: 0,
    Read onlys: 0, General errors: 0,
```

```
Total request varbinds: 0, Total set varbinds: 0,  
Get requests: 0, Get nexts: 0, Set requests: 0,  
Get responses: 0, Traps: 0,  
Silent drops: 0, Proxy drops 0  
Output:  
Packets: 0, Too bigs: 0, No such names: 0,  
Bad values: 0, General errors: 0,  
Get requests: 0, Get nexts: 0, Set requests: 0,  
Get responses: 0, Traps: 0
```

request snmp spoof-trap

Syntax	request snmp spoof-trap <trap> variable-bindings <object> <instance> <value>
Release Information	Command introduced in Junos OS Release 8.2. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Spoof (mimic) the behavior of a Simple Network Management Protocol (SNMP) trap.
Options	<p><trap>—Name of the trap to spoof.</p> <p>variable-bindings <object> <instance> <value>—(Optional) List of variables and values to include in the trap. Each variable binding is specified as an object name, the object instance, and the value (for example, ifIndex[14] = 14). Enclose the list of variable bindings in quotation marks (" ") and use a comma to separate each object name, instance, and value definition (for example, variable-bindings "ifIndex[14] = 14, ifAdminStatus[14] = 1, ifOperStatus[14] = 2"). Objects included in the trap definition that do not have instances and values specified as part of the command are included in the trap and spoofed with automatically generated instances and values.</p> <p><dummy name>—A dummy trap name to display the list of available traps.</p> <p>Question mark (?)—Question mark? to display possible completions.</p>
Required Privilege Level	request
List of Sample Output	request snmp spoof-trap (with Variable Bindings) on page 416 request snmp spoof-trap (Illegal Trap Name) on page 416 request snmp spoof-trap (Question Mark ?) on page 420

Sample Output

request snmp spoof-trap (with Variable Bindings)

```
user@host> request snmpspoof-trap linkUp variable-bindings "ifIndex[14] = 14, ifAdminStatus[14] = 1, ifOperStatus[14] = 2"
Spoof trap request result: trap sent successfully
```

request snmp spoof-trap (Illegal Trap Name)

```
user@host> request snmp spoof-trap xx
Spoof trap request result: trap not found
```

```
Allowed Traps:
ads1AtucInitFailureTrap
ads1AtucPerfESsThreshTrap
ads1AtucPerfLofsThreshTrap
ads1AtucPerfLolsThreshTrap
ads1AtucPerfLossThreshTrap
ads1AtucPerfLprsThreshTrap
ads1AtucRateChangeTrap
ads1AturPerfESsThreshTrap
```


adslAturPerfLossThreshTrap
adslAturPerfLossThreshTrap
adslAturPerfLprsThreshTrap
adslAturRateChangeTrap
apsEventChannelMismatch
apsEventFEPLF
apsEventModeMismatch
apsEventPSBF
apsEventSwitchover
authenticationFailure
bfdSessDown
bfdSessUp
bgpBackwardTransition
bgpEstablished
coldStart
dlswTrapCircuitDown
dlswTrapCircuitUp
dlswTrapTConnDown
dlswTrapTConnPartnerReject
dlswTrapTConnProtViolation
dlswTrapTConnUp
dsx1LineStatusChange
dsx3LineStatusChange
entConfigChange
fallingAlarm
frDLCIStatusChange
ggsnTrapChanged
ggsnTrapCleared
ggsnTrapNew
gmp1sTunnelDown
ifMauJabberTrap
ipv6IfStateChange
isisAreaMismatch
isisAttemptToExceedMaxSequence
isisAuthenticationFailure
isisAuthenticationTypeFailure
isisCorruptedLSPDetected
isisDatabaseOverload
isisIDLenMismatch
isisLSPTooLargeToPropagate
isisManualAddressDrops
isisMaxAreaAddressesMismatch
isisOriginatingLSPBufferSizeMismatch
isisOwnLSPPurge
isisProtocolsSupportedMismatch
isisRejectedAdjacency
isisSequenceNumberSkip
isisVersionSkew
jnxAccessAuthServerDisabled
jnxAccessAuthServerEnabled
jnxAccessAuthServiceDown
jnxAccessAuthServiceUp
jnxBfdSessDetectionTimeHigh
jnxBfdSessTxIntervalHigh
jnxBgpM2BackwardTransition
jnxBgpM2Established
jnxCmCfgChange
jnxCmRescueChange
jnxCollFlowOverload
jnxCollFlowOverloadCleared
jnxCollFtpSwitchover

jnxCollMemoryAvailable
jnxCollMemoryUnavailable
jnxCollUnavailableDest
jnxCollUnavailableDestCleared
jnxCollUnsuccessfulTransfer
jnxDfcHardMemThresholdExceeded
jnxDfcHardMemUnderThreshold
jnxDfcHardPpsThresholdExceeded
jnxDfcHardPpsUnderThreshold
jnxDfcSoftMemThresholdExceeded
jnxDfcSoftMemUnderThreshold
jnxDfcSoftPpsThresholdExceeded
jnxDfcSoftPpsUnderThreshold
jnxEventTrap
jnxExampleStartup
jnxFEBSwitchover
jnxFanFailure
jnxFanOK
jnxFruCheck
jnxFruFailed
jnxFruInsertion
jnxFruOK
jnxFruOffline
jnxFruOnline
jnxFruPowerOff
jnxFruPowerOn
jnxFruRemoval
jnxHardDiskFailed
jnxHardDiskMissing
jnxJsAvPatternUpdateTrap
jnxJsChassisClusterSwitchover
jnxJsFwAuthCapacityExceeded
jnxJsFwAuthFailure
jnxJsFwAuthServiceDown
jnxJsFwAuthServiceUp
jnxJsNatAddrPoolThresholdStatus
jnxJsScreenAttack
jnxJsScreenCfgChange
jnxLdpLspDown
jnxLdpLspUp
jnxLdpSesDown
jnxLdpSesUp
jnxMIMstCistPortLoopProtectStateChangeTrap
jnxMIMstCistPortRootProtectStateChangeTrap
jnxMIMstErrTrap
jnxMIMstGenTrap
jnxMIMstInvalidBpduRxdTrap
jnxMIMstMstiPortLoopProtectStateChangeTrap
jnxMIMstMstiPortRootProtectStateChangeTrap
jnxMIMstNewRootTrap
jnxMIMstProtocolMigrationTrap
jnxMIMstRegionConfigChangeTrap
jnxMIMstTopologyChgTrap
jnxMacChangedNotification
jnxMplsLdpInitSesThresholdExceeded
jnxMplsLdpPathVectorLimitMismatch
jnxMplsLdpSessionDown
jnxMplsLdpSessionUp
jnxOspfV3IfConfigError
jnxOspfV3IfRxBadPacket
jnxOspfV3IfStateChange

```

jnxOspfV3LsdbApproachingOverflow
jnxOspfV3LsdbOverflow
jnxOspfV3NbrRestartHelperStatusChange
jnxOspfV3NbrStateChange
jnxOspfV3NssaTranslatorStatusChange
jnxOspfV3RestartStatusChange
jnxOspfV3VirtIfConfigError
jnxOspfV3VirtIfRxBadPacket
jnxOspfV3VirtIfStateChange
jnxOspfV3VirtNbrRestartHelperStatusChange
jnxOspfV3VirtNbrStateChange
jnxOtnAlarmCleared
jnxOtnAlarmSet
jnxOverTemperature
jnxPMonOverloadCleared
jnxPMonOverloadSet
jnxPingEgressJitterThresholdExceeded
jnxPingEgressStdDevThresholdExceeded
jnxPingEgressThresholdExceeded
jnxPingIngressJitterThresholdExceeded
jnxPingIngressStdDevThresholdExceeded
jnxPingIngressThresholdExceeded
jnxPingRttJitterThresholdExceeded
jnxPingRttStdDevThresholdExceeded
jnxPingRttThresholdExceeded
jnxPortBpduErrorStatusChangeTrap
jnxPortLoopProtectStateChangeTrap
jnxPortRootProtectStateChangeTrap
jnxPowerSupplyFailure
jnxPowerSupplyOK
jnxRedundancySwitchover
jnxRmonAlarmGetFailure
jnxRmonGetOk
jnxSecAccessIfMacLimitExceeded
jnxSecAccessSdsRateLimitCrossed
jnxSonetAlarmCleared
jnxSonetAlarmSet
jnxSpSvcSetCpuExceeded
jnxSpSvcSetCpuOk
jnxSpSvcSetZoneEntered
jnxSpSvcSetZoneExited
jnxStormEventNotification
jnxSyslogTrap
jnxTemperatureOK
jnxVccpPortDown
jnxVccpPortUp
jnxVpnIfDown
jnxVpnIfUp
jnxVpnPwDown
jnxVpnPwUp
jnxl2aldGlobalMacLimit
jnxl2aldInterfaceMacLimit
jnxl2aldRoutingInstMacLimit
linkDown
linkUp
lldpRemTablesChange
mfrMibTrapBundleLinkMismatch
mplsLspChange
mplsLspDown
mplsLspInfoChange
mplsLspInfoDown

```

mplsLspInfoPathDown
mplsLspInfoPathUp
mplsLspInfoUp
mplsLspPathDown
mplsLspPathUp
mplsLspUp
mplsNumVrfRouteMaxThreshExceeded
mplsNumVrfRouteMidThreshExceeded
mplsNumVrfSecIllglLb1ThrshExcd
mplsTunnelDown
mplsTunnelReoptimized
mplsTunnelRerouted
mplsTunnelUp
mplsVrfIfDown
mplsVrfIfUp
mplsXCDown
mplsXCUp
msdpBackwardTransition
msdpEstablished
newRoot
ospfIfAuthFailure
ospfIfConfigError
ospfIfRxBadPacket
ospfIfStateChange
ospfLsdbApproachingOverflow
ospfLsdbOverflow
ospfMaxAgeLsa
ospfNbrStateChange
ospfOriginateLsa
ospfTxRetransmit
ospfVirtIfAuthFailure
ospfVirtIfConfigError
ospfVirtIfRxBadPacket
ospfVirtIfStateChange
ospfVirtIfTxRetransmit
ospfVirtNbrStateChange
pethMainPowerUsageOffNotification
pethMainPowerUsageOnNotification
pethPsePortOnOffNotification
pingProbeFailed
pingTestCompleted
pingTestFailed
ptopoConfigChange
risingAlarm
rpMauJabberTrap
sd1cLSStatusChange
sd1cPortStatusChange
topologyChange
traceRoutePathChange
traceRouteTestCompleted
traceRouteTestFailed
vrrpTrapAuthFailure
vrrpTrapNewMaster
warmStart

request snmp spoof-trap (Question Mark ?)

```
user@host> request snmp spoof-trap ?  
Possible completions:  
<trap>                The name of the trap to spoof  
ads1AtucInitFailureTrap
```

```
ads1AtucPerfESsThreshTrap
ads1AtucPerfLofsThreshTrap
ads1AtucPerfLolsThreshTrap
ads1AtucPerfLossThreshTrap
ads1AtucPerfLprsThreshTrap
ads1AtucRateChangeTrap
ads1AturPerfESsThreshTrap
ads1AturPerfLofsThreshTrap
ads1AturPerfLossThreshTrap
ads1AturPerfLprsThreshTrap
ads1AturRateChangeTrap
apsEventChannelMismatch
apsEventFEPLF
apsEventModeMismatch
apsEventPSBF
apsEventSwitchover
authenticationFailure
bfdSessDown
bfdSessUp
bgpBackwardTransition
bgpEstablished
coldStart
dlswTrapCircuitDown
dlswTrapCircuitUp
---(more 10%)---
```

show interfaces snmp-index

Syntax	<code>show interfaces snmp-index <i>snmp-index</i></code>
Release Information	Command introduced before Junos OS Release 7.4.
Description	Display information for the interface with the specified SNMP index.
Options	This command has no options.
Additional Information	Output from both the <code>show interfaces <i>interface-name</i> detail</code> and the <code>show interfaces <i>interface-name</i> extensive</code> command includes all the information displayed in the output from the <code>show interfaces snmp-index</code> command.
Required Privilege Level	view
List of Sample Output	show interfaces snmp-index on page 422
Output Fields	The output fields from the <code>show interfaces snmp-index <i>snmp-index</i></code> command are identical to those produced by the <code>show interfaces <i>interface-name</i></code> command. For a description of output fields, see the other chapters in this manual.

Sample Output

show interfaces snmp-index

```
user@host> show interfaces snmp-index 33
Physical interface: so-2/1/1, Enabled, Physical link is Down
  Interface index: 149, SNMP ifIndex: 33
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: 0C48,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down Point-To-Point SNMP-Traps 16384
  Link flags     : Keepalives
  CoS queues     : 8 supported
  Last flapped   : 2005-06-15 11:45:57 PDT (05:38:43 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : LOL, PLL, LOS
  SONET defects  : LOL, PLL, LOF, LOS, SEF, AIS-L, AIS-P
```

show ilmi statistics

Syntax	show ilmi statistics
Release Information	Command introduced before Junos OS Release 7.4.
Description	Display input and output Integrated Local Management Interface (ILMI) statistics.
Options	This command has no options.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• clear ilmi statistics on page 412
List of Sample Output	show ilmi statistics on page 425
Output Fields	Table 32 on page 424 lists the output fields for the show ilmi statistics command. Output fields are listed in the approximate order in which they appear.

Table 32: show ilmi statistics Output Fields

Field Name	Field Description
Input	<p>Information about received ILMI packets:</p> <ul style="list-style-type: none"> • Packets—Total number of messages delivered to the ILMI entity from the transport service. • Bad versions—Total number of messages delivered to the ILMI entity that were for an unsupported ILMI version. • Bad community names—Total number of messages delivered to the ILMI entity that did not use an ILMI community name. • Bad community uses—Total number of messages delivered to the ILMI entity that represented an ILMI operation that was not allowed by the ILMI community named in the message. • ASN parse errors—Total number of ASN.1 or BER errors encountered by the ILMI entity when decoding received ILMI messages. • Too bigs—Total number of ILMI packets delivered to the ILMI entity with an error status field of tooBig. • No such names—Total number of ILMI packets delivered to the ILMI entity with an error status field of noSuchName. • Bad values—Total number of ILMI packets delivered to the ILMI entity with an error status field of badValue. • Read onlys—Total number of valid ILMI packets delivered to the ILMI entity with an error status field of readOnly. Only incorrect implementations of ILMI generate this error. • General errors—Total number of ILMI packets delivered to the ILMI entity with an error status field of genErr. • Total request varbinds—Total number of objects retrieved successfully by the ILMI entity as a result of receiving valid ILMI GetRequest and GetNext packets. • Total set varbinds—Total number of objects modified successfully by the ILMI entity as a result of receiving valid ILMI SetRequest packets. • Get requests—Total number of ILMI GetRequest packets that have been accepted and processed by the ILMI entity. • Get nexts—Total number of ILMI GetNext packets that have been accepted and processed by the ILMI entity. • Set requests—Total number of ILMI SetRequest packets that have been accepted and processed by the ILMI entity. • Get responses—Total number of ILMI GetResponse packets that have been accepted and processed by the ILMI entity. • Traps—Total number of ILMI traps received by the ILMI entity. • Silent drops—Total number of GetRequest, GetNextRequest, GetBulkRequest, SetRequest, and InformRequest packets delivered to the ILMI entity that were silently dropped because the size of a reply containing an alternate response packet with an empty variable-bindings field was greater than either a local constraint or the maximum message size associated with the originator of the requests. • Proxy drops—Total number of GetRequest, GetNextRequest, GetBulkRequest, SetRequest, and InformRequest packets delivered to the ILMI entity that were silently dropped because the transmission of the (possibly translated) message to a proxy target failed in such a way (other than a timeout) that no response packet could be returned.
Output	<p>Information about transmitted ILMI packets:</p> <ul style="list-style-type: none"> • Packets—Total number of messages passed from the ILMI entity to the transport service. • Too bigs—Total number of ILMI packets generated by the ILMI entity with an error status field of tooBig. • No such names—Total number of ILMI packets generated by the ILMI entity with an error status field of noSuchName. • Bad values—Total number of ILMI packets generated by the ILMI entity with an error status field of badValue. • General errors—Total number of ILMI packets generated by the ILMI entity with an error status field of genErr. • Get requests—Total number of ILMI GetRequest packets that have been generated by the ILMI entity. • Get nexts—Total number of ILMI GetNext packets that have been generated by the ILMI entity. • Set requests—Total number of ILMI SetRequest packets that have been generated by the ILMI entity. • Get responses—Total number of ILMI GetResponse packets that have been generated by the ILMI entity. • Traps—Total number of ILMI traps generated by the ILMI entity.

Sample Output

show ilmi statistics

```
user@host> show ilmi statistics
ILMI statistics:
  Input:
    Packets: 0, Bad versions: 0, Bad community names: 0,
    Bad community uses: 0, ASN parse errors: 0,
    Too bigs: 0, No such names: 0, Bad values: 0,
    Read onlys: 0, General errors: 0,
    Total request varbinds: 0, Total set varbinds: 0,
    Get requests: 0, Get nexts: 0, Set requests: 0,
    Get responses: 0, Traps: 0,
    Silent drops: 0, Proxy drops 0
  Output:
    Packets: 0, Too bigs: 0, No such names: 0,
    Bad values: 0, General errors: 0,
    Get requests: 0, Get nexts: 0, Set requests: 0,
    Get responses: 0, Traps: 0
```

show snmp health-monitor

Syntax	<code>show snmp health-monitor</code> <code><alarms <detail>> <logs></code>
Release Information	Command introduced in Junos OS Release 8.0. Command introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series.
Description	Display information about Simple Network Management Protocol (SNMP) health monitor alarms and logs.
Options	none —Display information about all health monitor alarms and logs. alarms <detail> —(Optional) Display detailed information about health monitor alarms. logs —(Optional) Display information about health monitor logs.
Required Privilege Level	view
List of Sample Output	show snmp health-monitor on page 428 show snmp health-monitor alarms detail on page 430
Output Fields	Table 33 on page 426 describes the output fields for the show snmp health-monitor command. Output fields are listed in the approximate order in which they appear.

Table 33: show snmp health-monitor Output Fields

Field Name	Field Description	Level of Output
Alarm Index	Alarm identifier.	All levels
Variable description	Description of the health monitor object instance being monitored.	All levels
Variable name	Name of the health monitor object instance being monitored.	All levels
Value	Current value of the monitored variable in the most recent sample interval.	All levels

Table 33: show snmp health-monitor Output Fields (*continued*)

Field Name	Field Description	Level of Output
State	<p>State of the alarm or event entry:</p> <ul style="list-style-type: none"> Alarms: <ul style="list-style-type: none"> active—Entry is fully configured and activated. falling threshold crossed—Value of the variable has crossed the lower threshold limit. rising threshold crossed—Value of the variable has crossed the upper threshold limit. under creation—Entry is being configured and is not yet activated. startup—Alarm is waiting for the first sample of the monitored variable. object not available—Monitored variable of that type is not available to the health monitor agent. instance not available—Monitored variable's instance is not available to the health monitor agent. object type invalid—Monitored variable is not a numeric value. object processing errored—An error occurred when the monitored variable was processed. unknown—State is not one of the above. 	All levels
Variable OID	Object ID to which the variable name is resolved. The format is x.x.x.x.	detail
Sample type	Method of sampling the monitored variable and calculating the value to compare against the upper and lower thresholds. It can have the value of absolute value or delta value .	detail
Startup alarm	<p>Alarm that might be sent when this entry is first activated, depending on the following criteria:</p> <ul style="list-style-type: none"> Alarm is sent when one of the following situations exists: <ul style="list-style-type: none"> Value of the alarm is above or equal to the rising threshold and the startup type is either rising alarm or rising or falling alarm. Value of the alarm is below or equal to the falling threshold and the startup type is either falling alarm or rising or falling alarm. Alarm is <i>not</i> sent when one of the following situations exists: <ul style="list-style-type: none"> Value of the alarm is above or equal to the rising threshold and the startup type is falling alarm. Value of the alarm is below or equal to the falling threshold and the startup type is rising alarm. Value of the alarm is between the thresholds. 	detail
Owner	Name of the entry configured by the user. If the entry was created through the CLI, the owner has monitor prepended to it.	detail
Creator	Mechanism by which the entry was configured (Health Monitor).	detail
Sample interval	Time period between samples (in seconds).	detail
Rising threshold	Upper limit threshold value as a percentage of the maximum possible value.	detail

Table 33: show snmp health-monitor Output Fields (*continued*)

Field Name	Field Description	Level of Output
Falling threshold	Lower limit threshold value as a percentage of the maximum possible value.	detail
Rising event index	Event triggered when the rising threshold is crossed.	detail
Falling event index	Event triggered when the falling threshold is crossed.	detail

Sample Output

show snmp health-monitor

```
user@host> show snmp health-monitor
```

```

Alarm
Index  Variable description                                Value State

32768 Health Monitor: root file system utilization
      jnxHrStoragePercentUsed.1                        58 active

32769 Health Monitor: /config file system utilization
      jnxHrStoragePercentUsed.2                        0 active

32770 Health Monitor: RE 0 CPU utilization
      jnxOperatingCPU.9.1.0.0                          0 active

32773 Health Monitor: RE 0 Memory utilization
      jnxOperatingBuffer.9.1.0.0                      35 active

32775 Health Monitor: jkernel daemon CPU utilization
      Init daemon                                     0 active
      Chassis daemon                                  50 active
      Firewall daemon                                 0 active
      Interface daemon                                5 active
      SNMP daemon                                     11 active
      MIB2 daemon                                     42 active
      Sonet APS daemon                                0 active
      VRRP daemon                                     0 active
      Alarm daemon                                    3 active
      PFE daemon                                      0 active
      CRAFT daemon                                    0 active
      Traffic sampling control daemon                 0 active
      Ilmi daemon                                     0 active
      Remote operations daemon                       0 active
      CoS daemon                                      0 active
      Pic Services Logging daemon                    0 active
      Internal Routing Service Daemon                 3 active
      Network Access Service daemon                  0 active
      Forwarding UDP daemon                          0 active
      Routing socket proxy daemon                    0 active
      Disk Monitoring daemon                         1 active
      Inet daemon                                     0 active
      Syslog daemon                                   0 active
      Adaptive Services PIC daemon                   0 active
      ECC parity errors logging Daemon                0 active
      Layer 2 Tunneling Protocol daemon               0 active
      PPPoE daemon                                    3 active

```

Redundancy device daemon	0 active
PPP daemon	0 active
Dynamic Flow Capture Daemon	0 active
32776 Health Monitor: jroute daemon CPU utilization	
Routing protocol daemon	1 active
Management daemon	0 active
Management daemon	0 active
Command line interface	4 active
Periodic Packet Management daemon	0 active
Link Management daemon	0 active
Pragmatic General Multicast daemon	0 active
Bidirectional Forwarding Detection daemon	0 active
SRC daemon	0 active
audit daemon	0 active
Event daemon	0 active
32777 Health Monitor: jcrypto daemon CPU utilization	
IPSec Key Management daemon	0 active
32779 Health Monitor: jkernel daemon Memory utilization	
Init daemon	47384 active
Chassis daemon	20204 active
Firewall daemon	1956 active
Interface daemon	3340 active
SNMP daemon	4540 active
MIB2 daemon	3880 active
Sonet APS daemon	2632 active
VRRP daemon	2672 active
Alarm daemon	1856 active
PFE daemon	2600 active
CRAFT daemon	2000 active
Traffic sampling control daemon	3164 active
Ilmi daemon	2132 active
Remote operations daemon	2964 active
CoS daemon	3044 active
Pic Services Logging daemon	1944 active
Internal Routing Service Daemon	1392 active
Network Access Service daemon	1992 active
Forwarding UDP daemon	1876 active
Routing socket proxy daemon	1296 active
Disk Monitoring daemon	1180 active
Inet daemon	1296 active
Syslog daemon	1180 active
Adaptive Services PIC daemon	3220 active
ECC parity errors logging Daemon	1100 active
Layer 2 Tunneling Protocol daemon	3372 active
PPPoE daemon	1424 active
Redundancy device daemon	1820 active
PPP daemon	2060 active
Dynamic Flow Capture Daemon	10740 active
32780 Health Monitor: jroute daemon Memory utilization	
Routing protocol daemon	8104 active
Management daemon	13360 active
Management daemon	19252 active
Command line interface	9912 active
Periodic Packet Management daemon	1484 active
Link Management daemon	2016 active
Pragmatic General Multicast daemon	1968 active
Bidirectional Forwarding Detection daemon	1956 active
SRC daemon	1772 active

audit daemon	1772 active
Event daemon	1808 active

32781 Health Monitor: jcrypto daemon Memory utilization	
IPSec Key Management daemon	5600 active

show snmp health-monitor alarms detail

```
user@host> show snmp health-monitor alarms detail
```

```
Alarm Index 32768:
  Variable name      jnxHrStoragePercentUsed.1
  Variable OID       1.3.6.1.4.1.2636.3.31.1.1.1.1.1
  Sample type        absolute value
  Startup alarm       rising alarm
  Owner               Health Monitor: root file system
                     utilization
  Creator             Health Monitor
  State               active
  Sample interval     300 seconds
  Rising threshold    80
  Falling threshold   70
  Rising event index  32768
  Falling event index 32768
  Instance Value: 58
  Instance State: active

Alarm Index 32769:
  Variable name      jnxHrStoragePercentUsed.2
  Variable OID       1.3.6.1.4.1.2636.3.31.1.1.1.1.2
  Sample type        absolute value
  Startup alarm       rising alarm
  Owner               Health Monitor: /config file system
                     utilization
  Creator             Health Monitor
  State               active
  Sample interval     300 seconds
  Rising threshold    80
  Falling threshold   70
  Rising event index  32768
  Falling event index 32768
  Instance Value: 0
  Instance State: active

Alarm Index 32770:
  Variable name      jnxOperatingCPU.9.1.0.0
  Variable OID       1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0
  Sample type        absolute value
  Startup alarm       rising alarm
  Owner               Health Monitor: RE 0 CPU utilization

  Creator             Health Monitor
  State               active
  Sample interval     300 seconds
  Rising threshold    80
  Falling threshold   70
  Rising event index  32768
  Falling event index 32768
  Instance Value: 0
  Instance State: active
```

Alarm Index 32773:

Variable name jnxOperatingBuffer.9.1.0.0
 Variable OID 1.3.6.1.4.1.2636.3.1.13.1.11.9.1.0.0
 Sample type absolute value
 Startup alarm rising alarm
 Owner Health Monitor: RE 0 Memory utilization

Creator Health Monitor
 State active
 Sample interval 300 seconds
 Rising threshold 80
 Falling threshold 70
 Rising event index 32768
 Falling event index 32768
 Instance Value: 35
 Instance State: active

Alarm Index 32775:

Variable name sysAppElmtRunCPU.3
 Variable OID 1.3.6.1.2.1.54.1.2.3.1.9.3
 Sample type delta value
 Startup alarm rising alarm
 Owner Health Monitor: jkernel daemon CPU utilization

Creator Health Monitor
 State active
 Sample interval 300 seconds
 Rising threshold 24000
 Falling threshold 21000
 Rising event index 32768
 Falling event index 32768
 Instance Name: sysAppElmtRunCPU.3.1.1
 Instance Description: Init daemon
 Instance Value: 0
 Instance State: active

Instance Name: sysAppElmtRunCPU.3.2.2786
 Instance Description: Chassis daemon
 Instance Value: 50
 Instance State: active

Instance Name: sysAppElmtRunCPU.3.3.2938
 Instance Description: Firewall daemon
 Instance Value: 0
 Instance State: active

Instance Name: sysAppElmtRunCPU.3.4.2942
 Instance Description: Interface daemon
 Instance Value: 5
 Instance State: active

Instance Name: sysAppElmtRunCPU.3.7.7332
 Instance Description: SNMP daemon
 Instance Value: 11
 Instance State: active

Instance Name: sysAppElmtRunCPU.3.9.2914
 Instance Description: MIB2 daemon
 Instance Value: 42

```
Instance State: active

Instance Name: sysAppElemRunCPU.3.12.2916
Instance Description: Sonet APS daemon
Instance Value: 0
Instance State: active

Instance Name: sysAppElemRunCPU.3.13.2917
Instance Description: VRRP daemon
Instance Value: 0
Instance State: active

Instance Name: sysAppElemRunCPU.3.14.2787
Instance Description: Alarm daemon
Instance Value: 3
Instance State: active

Instance Name: sysAppElemRunCPU.3.15.2940
Instance Description: PFE daemon
Instance Value: 0
Instance State: active

Instance Name: sysAppElemRunCPU.3.16.2788
Instance Description: CRAFT daemon
Instance Value: 0
Instance State: active

Instance Name: sysAppElemRunCPU.3.17.2918
Instance Description: Traffic sampling control daemon
---(more 23%)---
```


show snmp inform-statistics

Syntax	show snmp inform-statistics
Release Information	Command introduced in Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Display information about Simple Network Management Protocol (SNMP) inform requests.
Options	This command has no options.
Required Privilege Level	view
List of Sample Output	show snmp inform-statistics on page 433
Output Fields	Table 34 on page 433 describes the output fields for the show snmp inform-statistics command. Output fields are listed in the approximate order in which they appear.

Table 34: show snmp inform-statistics Output Fields

Field Name	Field Description
Target Name	Name of the device configured to receive and respond to SNMP informs.
Address	IP address of the target device.
Sent	Number of informs sent to the target device and acknowledged by the target device.
Pending	Number of informs held in memory pending a response from the target device.
Discarded	Number of informs discarded after the specified number of retransmissions to the target device were attempted.
Timeouts	Number of informs that did not receive an acknowledgement from the target device within the timeout specified.
Probe Failures	Connection failures that occurred (for example, when the target server returned invalid content or you incorrectly configured the target address).

Sample Output

show snmp inform-statistics

```

user@host> show snmp inform-statistics
Inform Request Statistics:
  Target Name: TA1_v3_md5_none Address: 172.17.20.184
    Sent: 176, Pending: 0
    Discarded: 0, Timeouts: 0, Probe Failures: 0
  Target Name: TA2_v3_sha_none Address: 192.168.110.59

```

Sent: 0, Pending: 4
Discarded: 84, Timeouts: 0, Probe Failures: 258
Target Name: TA5_v2_none Address: 172.17.20.184
Sent: 0, Pending: 0
Discarded: 2, Timeouts: 10, Probe Failures: 0

show snmp mib

Syntax	<code>show snmp mib (get get-next walk) (ascii decimal) <i>object-id</i></code>
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>ascii and decimal options introduced in Junos OS Release 9.6.</p> <p>ascii and decimal options introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Command introduced in Junos OS Release 11.1 for the QFX Series.</p>
Description	Display local Simple Network Management Protocol (SNMP) Management Information Base (MIB) object values.
Options	<p>get—Retrieve and display one or more SNMP object values.</p> <p>get-next—Retrieve and display the next SNMP object values.</p> <p>walk—Retrieve and display the SNMP object values that are associated with the requested object identifier (OID). When you use this option, the Junos OS displays the objects below the subtree that you specify.</p> <p>ascii—Display the SNMP object's string indices as an ASCII-key representation.</p> <p>decimal—Display the SNMP object values in the decimal (default) format. The decimal option is the default option for this command. Therefore, issuing the show snmp mib (get get-next walk) decimal object-id and the show snmp mib (get get-next walk) object-id commands display the same output.</p> <p>object-id—The object can be represented by a sequence of dotted integers (such as 1.3.6.1.2.1.2) or by its subtree name (such as interfaces). When entering multiple objects, enclose the objects in quotation marks.</p>
Required Privilege Level	snmp—To view this statement in the configuration.
List of Sample Output	<p>show snmp mib get on page 436</p> <p>show snmp mib get (Multiple Objects) on page 436</p> <p>show snmp mib get (Layer 2 Policer) on page 436</p> <p>show snmp mib get-next on page 436</p> <p>show snmp mib get-next (Specify an OID) on page 436</p> <p>show snmp mib walk on page 436</p> <p>show snmp mib walk (QFX Series) on page 436</p> <p>show snmp mib walk decimal on page 437</p> <p>show snmp mib walk (ASCII) on page 437</p> <p>show snmp mib walk (Multiple Indices) on page 437</p> <p>show snmp mib walk decimal (Multiple Indices) on page 437</p>
Output Fields	Table 35 on page 436 describes the output fields for the show snmp mib command. Output fields are listed in the approximate order in which they appear.

Table 35: show snmp mib Output Fields

Field Name	Field Description
<i>name</i>	Object name and numeric instance value.
<i>object value</i>	Object value. The Junos OS translates OIDs into the corresponding object names.

Sample Output

show snmp mib get

```
user@host> show snmp mib get sysObjectID.0
sysObjectID.0 = jnxProductNameM20
```

show snmp mib get (Multiple Objects)

```
user@host> show snmp mib get ?sysObjectID.0 sysUpTime.0?
sysObjectID.0 = jnxProductNameM20
sysUpTime.0 = 1640992
```

show snmp mib get (Layer 2 Policer)

```
user@host> show snmp mib get ifInOctets.25970
ifInOctets.25970 = 7545720
```

show snmp mib get-next

```
user@host> show snmp mib get-next jnxMibs
jnxBoxClass.0 = jnxProductLineM20.0
```

show snmp mib get-next (Specify an OID)

```
user@host> show snmp mib get-next 1.3.6.1
sysDescr.0 = Juniper Networks, Inc. m20 internet router, kernel
Junos OS Release: 2004-1 Build date: build date UTC Copyright (c) 1996-2004 Juniper
Networks, Inc.
```

show snmp mib walk

```
user@host> show snmp mib walk system
sysDescr.0 = Juniper Networks, Inc. m20 internet router, kernel
Junos OS Release #0: 2004-1 Build date: build date UTC Copyright (c) 1996-2004
Juniper Networks, Inc.
sysObjectID.0 = jnxProductNameM20
sysUpTime.0 = 1640992
sysContact.0 = Your contact
sysName.0 = my router
sysLocation.0 = building 1
sysServices.0 = 4
```

show snmp mib walk (QFX Series)

```
user@switch> show snmp mib walk system
sysDescr.0 = Juniper Networks, Inc. qfx3500s internet router, kernel JUNOS
11.1-20100926.0 #0: 2010-09-26 06:17:38 UTC Build date: 2010-09-26 06:00:10
sysObjectID.0 = jnxProductQFX3500
sysUpTime.0 = 138980301
sysContact.0 = System Contact
```

```
sysName.0      = LabQFX3500
sysLocation.0 = Lab
sysServices.0 = 4
```

show snmp mib walk decimal

```
user@host show snmp mib walk decimal jnxUtilData
jnxUtilCounter32Value.102.114.101.100 = 100
```

show snmp mib walk (ASCII)

```
show snmp mib walk ascii jnxUtilData
jnxUtilCounter32Value."fred" = 100
```

show snmp mib walk (Multiple Indices)

```
show snmp mib walk ascii jnxFWCounterByteCount
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_BE-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_CC-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_RT-fe-1/3/0.0-i".2 = 0
.....
```

show snmp mib walk decimal (Multiple Indices)

```
show snmp mib walk decimal jnxFWCounterByteCount
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_BE-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_CC-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_RT-fe-1/3/0.0-i".2 = 0
.....
```

show snmp rmon

Syntax	<code>show snmp rmon</code> <code><alarms <brief detail> events <brief detail> logs></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display information about Simple Network Management Protocol (SNMP) Remote Monitoring (RMON) alarms and events.
Options	<p>none—Display information about all RMON alarms and events.</p> <p>alarms—(Optional) Display information about RMON alarms.</p> <p>brief detail—(Optional) Display brief or detailed information about RMON alarms or events.</p> <p>events—(Optional) Display information about RMON events.</p> <p>logs—(Optional) Display information about RMON monitoring logs.</p>
Required Privilege Level	view
List of Sample Output	<p>show snmp rmon on page 440</p> <p>show snmp rmon alarms detail on page 440</p> <p>show snmp rmon events detail on page 441</p>
Output Fields	Table 36 on page 438 describes the output fields for the show snmp rmon command. Output fields are listed in the approximate order in which they appear.

Table 36: show snmp rmon Output Fields

Field Name	Field Description	Level of Output
Alarm Index	Alarm identifier.	All levels

Table 36: show snmp rmon Output Fields (*continued*)

Field Name	Field Description	Level of Output
State	<p>State of the alarm or event entry:</p> <p>Alarms:</p> <ul style="list-style-type: none"> • active—Entry is fully configured and activated. • falling threshold crossed—Value of the variable has crossed the lower threshold limit. • rising threshold crossed—Value of the variable has crossed the upper threshold limit. • under creation—Entry is being configured and is not yet activated. • startup—Alarm is waiting for the first sample of the monitored variable. • object not available—Monitored variable of that type is not available to the SNMP agent. • instance not available—Monitored variable's instance is not available to the SNMP agent. • object type invalid—Monitored variable is not a numeric value. • object processing errored—An error occurred when the monitored variable was processed. • unknown—State is not one of the above. <p>Events:</p> <ul style="list-style-type: none"> • active—Entry has been fully configured and activated. • under creation—Entry is being configured and is not yet activated. • unknown—State is not one of the above. 	All levels
Variable name	Name of the SNMP object instance being monitored.	All levels
Event Index	Event identifier.	All levels
Type	<p>Type of notification made when an event is triggered. It can be one of the following:</p> <ul style="list-style-type: none"> • log—A system log message is generated and an entry is made to the log table. • snmptrap—An SNMP trap is sent to the configured destination. • log and trap—A system log message is generated, an entry is made to the log table, and an SNMP trap is sent to the configured destination. • none—Neither log nor trap will be sent. 	detail
Last Event	Date and time of the last event. It has the format <i>yyyy-mm-dd hh:mm:ss timezone</i> .	brief
Community	Identifies the trap group used for sending the SNMP trap.	detail
Variable OID	Object ID to which the variable name is resolved. The format is x.x.x.x.	detail
Sample type	Method of sampling the monitored variable and calculating the value to compare against the upper and lower thresholds. It can have the value of absolute value or delta value .	detail

Table 36: show snmp rmon Output Fields (*continued*)

Field Name	Field Description	Level of Output
Startup alarm	Alarm that might be sent when this entry is first activated, depending on the following criteria: <ul style="list-style-type: none"> Alarm is sent when one of the following situations exists: <ul style="list-style-type: none"> Value of the alarm is above or equal to the rising threshold and the startup type is either rising alarm or rising or falling alarm. Value of the alarm is below or equal to the falling threshold and the startup type is either falling alarm or rising or falling alarm. Alarm is <i>not</i> sent when one of the following situations exists: <ul style="list-style-type: none"> Value of the alarm is above or equal to the rising threshold and the startup type is falling alarm. Value of the alarm is below or equal to the falling threshold and the startup type is rising alarm. Value of the alarm is between the thresholds. 	detail
Owner	Name of the entry configured by the user. If the entry was created through the CLI, the owner has monitor prepended to it.	detail
Creator	Mechanism by which the entry was configured (CLI or SNMP).	detail
Sample interval	Time period between samples (in seconds).	detail
Rising threshold	Upper limit threshold value configured by the user.	detail
Falling threshold	Lower limit threshold value configured by the user.	detail
Rising event index	Event triggered when the rising threshold is crossed.	detail
Falling event index	Event triggered when the falling threshold is crossed.	detail
Current value	Current value of the monitored variable in the most recent sample interval.	detail

Sample Output

show snmp rmon

```

user@host> show snmp rmon
Alarm
Index  State                               Variable name
   1   falling threshold crossed         ifInOctets.1

Event
Index  Type                                Last Event
   1   log and trap                      2002-01-30 01:13:01 PST

```

show snmp rmon alarms detail

```

user@host> show snmp rmon alarms detail

```



```
Alarm Index 1:
Variable name          ifInOctets.1
Variable OID           1.3.6.1.2.1.2.2.1.10.1
Sample type            delta value
Startup alarm          rising or falling alarm
Owner                  monitor
Creator                CLI
State                  falling threshold crossed
Sample interval        60 seconds
Rising threshold       100000
Falling threshold      80000
Rising event index     1
Falling event index    1
Current value          0
```

show snmp rmon events detail

```
user@host> show snmp rmon events detail
Event Index 1:
Type                log and trap
Community            boy-elroy
Last event           2002-01-30 01:13:01 PST
Creator              CLI
State                active
```

show snmp statistics

Syntax	show snmp statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Display statistics about Simple Network Management Protocol (SNMP) packets sent and received by the router or switch.
Options	This command has no options.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • clear snmp statistics on page 414
List of Sample Output	show snmp statistics on page 445
Output Fields	Table 37 on page 442 describes the output fields for the show snmp statistics command. Output fields are listed in the approximate order in which they appear.

Table 37: show snmp statistics Output Fields

Field Name	Field Description
Input	<p>Information about received packets:</p> <ul style="list-style-type: none"> • Packets(snmplnPkts)—Total number of messages delivered to the SNMP entity from the transport service. • Bad versions—(snmplnBadVersions) Total number of messages delivered to the SNMP entity that were for an unsupported SNMP version. • Bad community names—(snmplnBadCommunityNames) Total number of messages delivered to the SNMP entity that used an SNMP community name not known to the entity. • Bad community uses—(snmplnBadCommunityUses) Total number of messages delivered to the SNMP entity that represented an SNMP operation that was not allowed by the SNMP community named in the message. • ASN parse errors—(snmplnASNParseErrs) Total number of ASN.1 or BER errors encountered by the SNMP entity when decoding received SNMP messages. • Too big—(snmplnTooBig) Total number of SNMP PDUs delivered to the SNMP entity with an error status field of tooBig. • No such names—(snmplnNoSuchNames) Total number of SNMP PDUs delivered to the SNMP entity with an error status field of noSuchName. • Bad values—(snmplnBadValues) Total number of SNMP PDUs delivered to the SNMP entity with an error status field of badValue. • Read only—(snmplnReadOnly) Total number of valid SNMP PDUs delivered to the SNMP entity with an error status field of readOnly. Only incorrect implementations of SNMP generate this error.

Table 37: show snmp statistics Output Fields (*continued*)

Field Name	Field Description
Input (continued)	<ul style="list-style-type: none"> • General errors—(snmpInGenErrs) Total number of SNMP PDUs delivered to the SNMP entity with an error status field of genErr. • Total requests varbinds—(snmpInTotalReqVars) Total number of MIB objects retrieved successfully by the SNMP entity as a result of receiving valid SNMP GetRequest and GetNext PDUs. • Total set varbinds—(snmpInSetVars) Total number of MIB objects modified successfully by the SNMP entity as a result of receiving valid SNMP SetRequest PDUs. • Get requests—(snmpInGetRequests) Total number of SNMP GetRequest PDUs that have been accepted and processed by the SNMP entity. • Get nexts—(snmpInGetNexts) Total number of SNMP GetNext PDUs that have been accepted and processed by the SNMP entity. • Set requests—(snmpInSetRequests) Total number of SNMP SetRequest PDUs that have been accepted and processed by the SNMP entity. • Get responses—(snmpInGetResponses) Total number of SNMP GetResponse PDUs that have been accepted and processed by the SNMP entity. • Traps—(snmpInTraps) Total number of SNMP traps generated by the SNMP entity. • Silent drops—(snmpSilentDrops) Total number of GetRequest, GetNextRequest, GetBulkRequest, SetRequests, and InformRequest PDUs delivered to the SNMP entity that were silently dropped because the size of a reply containing an alternate response PDU with an empty variable-bindings field was greater than either a local constraint or the maximum message size associated with the originator of the requests. • Proxy drops—(snmpProxyDrops) Total number of GetRequest, GetNextRequest, GetBulkRequest, SetRequests, and InformRequest PDUs delivered to the SNMP entity that were silently dropped because the transmission of the message to a proxy target failed in such a way (other than a timeout) that no response PDU could be returned. • Commit pending drops—Number of SNMP packets for Set requests dropped because of a previous pending SNMP Set request on the committed configuration. • Throttle drops—Number of SNMP packets for any requests dropped reaching the throttle limit.

Table 37: show snmp statistics Output Fields (*continued*)

Field Name	Field Description
V3 Input	<p>Information about SNMP version 3 packets:</p> <ul style="list-style-type: none"> • Unknown security models—(snmpUnknownSecurityModels) Total number of packets received by the SNMP engine that were dropped because they referenced a security model that was not known to or supported by the SNMP engine. • Invalid messages—(snmpInvalidMsgs) Number of packets received by the SNMP engine that were dropped because there were invalid or inconsistent components in the SNMP message. • Unknown pdu handlers—(snmpUnknownPDUHandlers) Number of packets received by the SNMP engine that were dropped because the PDU contained in the packet could not be passed to an application responsible for handling the PDU type. • Unavailable contexts—(snmpUnavailableContexts) Number of requests received for a context that is known to the SNMP engine, but is currently unavailable. • Unknown contexts—(snmpUnknownContexts) Total number of requests received for a context that is unknown to the SNMP engine. • Unsupported security levels—(usmStatsUnsupportedSecLevels) Total number of packets received by the SNMP engine that were dropped because they requested a security level unknown to the SNMP engine (or otherwise unavailable). • Not in time windows—(usmStatsNotInTimeWindows) Total number of packets received by the SNMP engine that were dropped because they appeared outside the authoritative SNMP engine's window. • Unknown user names—(usmStatsUnknownUserNames) Total number of packets received by the SNMP engine that were dropped because they referenced a user that was not known to the SNMP engine. • Unknown engine ids—(usmStatsUnknownEngineIDs) Total number of packets received by the SNMP engine that were dropped because they referenced an SNMP engine ID that was not known to the SNMP engine. • Wrong digests—(usmStatsWrongDigests) Total number of packets received by the SNMP engine that were dropped because they did not contain the expected digest value. • Decryption errors—(usmStatsDecryptionErrors) Total number of packets received by the SNMP engine that were dropped because they could not be decrypted.

Table 37: show snmp statistics Output Fields (*continued*)

Field Name	Field Description
Output	<p>Information about transmitted packets:</p> <ul style="list-style-type: none"> • Packets—(snmpOutPkts) Total number of messages passed from the SNMP entity to the transport service. • Too big—(snmpOutTooBig) Total number of SNMP PDUs generated by the SNMP entity with an error status field of tooBig. • No such names—(snmpOutNoSuchNames) Total number of SNMP PDUs delivered to the SNMP entity with an error status field of noSuchName. • Bad values—(snmpOutBadValues) Total number of SNMP PDUs generated by the SNMP entity with an error status field of badValue. • General errors—(snmpOutGenErrs) Total number of SNMP PDUs generated by the SNMP entity with an error status field of genErr. • Get requests—(snmpOutGetRequests) Total number of SNMP GetRequest PDUs generated by the SNMP entity. • Get nexts—(snmpOutGetNexts) Total number of SNMP GetNext PDUs generated by the SNMP entity. • Set requests—(snmpOutSetRequests) Total number of SNMP SetRequest PDUs generated by the SNMP entity. • Get responses—(snmpOutGetResponses) Total number of SNMP GetResponse PDUs generated by the SNMP entity. • Traps—(snmpOutTraps) Total number of SNMP traps generated by the SNMP entity.

Sample Output

show snmp statistics

```

user@host> show snmp statistics
SNMP statistics:
  Input:
    Packets: 246213, Bad versions: 12, Bad community names: 12,
    Bad community uses: 0, ASN parse errors: 96,
    Too big: 0, No such names: 0, Bad values: 0,
    Read onlys: 0, General errors: 0,
    Total request varbinds: 227084, Total set varbinds: 67,
    Get requests: 44942, Get nexts: 190371, Set requests: 10712,
    Get responses: 0, Traps: 0,
    Silent drops: 0, Proxy drops: 0, Commit pending drops: 0,
    Throttle drops: 0,
  V3 Input:
    Unknown security models: 0, Invalid messages: 0
    Unknown pdu handlers: 0, Unavailable contexts: 0
    Unknown contexts: 0, Unsupported security levels: 1
    Not in time windows: 0, Unknown user names: 0
    Unknown engine ids: 44, Wrong digests: 23, Decryption errors: 0
  Output:
    Packets: 246093, Too big: 0, No such names: 31561,
    Bad values: 0, General errors: 2,
    Get requests: 0, Get nexts: 0, Set requests: 0,
    Get responses: 246025, Traps: 0

```

show snmp v3

Syntax	<code>show snmp v3</code> <code><access <brief detail> community general groups notify <filter> target <address parameters> users></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display the Simple Network Management Protocol version 3 (SNMPv3) operating configuration.
Options	<p>none—Display all of the SNMPv3 operating configuration.</p> <p>access—(Optional) Display SNMPv3 access information.</p> <p>brief detail—(Optional) Display brief or detailed information about SNMPv3 access information.</p> <p>community—(Optional) Display SNMPv3 community information.</p> <p>general—(Optional) Display SNMPv3 general information.</p> <p>groups—(Optional) Display SNMPv3 security-to-group information.</p> <p>notify <filter>—(Optional) Display SNMPv3 notify and, optionally, notify filter information.</p> <p>target <address parameters>—(Optional) Display SNMPv3 target and, optionally, either target address or target parameter information.</p> <p>users—(Optional) Display SNMPv3 user information.</p>
Additional Information	To edit the default display of the show snmp v3 command, specify options in the show statement at the [edit snmp v3] hierarchy level.
Required Privilege Level	view
List of Sample Output	show snmp v3 on page 448
Output Fields	Table 38 on page 447 describes the output fields for the show snmp v3 command. Output fields are listed in the approximate order in which they appear.

Table 38: show snmp v3 Output Fields

Field Name	Field Description
Access control	<p>Information about access control:</p> <ul style="list-style-type: none"> • Group—Group name for which the configured access privileges apply. The group, together with the context prefix and the security model and security level, forms the index for this table. • Context prefix—SNMPv3 context for which the configured access privileges apply. • Security model/level—Security model and security level for which the configuration access privileges apply. • Read view—Identifies the MIB view applied to SNMPv3 read operations. • Write view—Identifies the MIB view applied to SNMPv3 write operations. • Notify view—Identifies the MIB view applied to outbound SNMP notifications.
Engine	<p>Information about local engine configuration:</p> <ul style="list-style-type: none"> • Local engine ID—Identifier that uniquely and unambiguously identifies the local SNMPv3 engine. • Engine boots—Number of times the local SNMPv3 engine has rebooted or reinitialized since the engine ID was last changed. • Engine time—Number of seconds since the local SNMPv3 engine was last rebooted or reinitialized. • Max msg size—Maximum message size the sender can accommodate.
Engine ID	<p>Information about engine ID:</p> <ul style="list-style-type: none"> • Local engine ID—Identifier that uniquely and unambiguously identifies the local SNMPv3 engine. • Engine boots—Number of times the local SNMPv3 engine has rebooted or reinitialized since the engine ID was last changed. • Engine time—Number of seconds since the local SNMPv3 engine was last rebooted or reinitialized. • Max msg size—Maximum message size the sender can accommodate. • Engine ID—SNMPv3 engine ID associated with each user. • User—SNMPv3 user. • Auth/Priv—Authentication and encryption algorithm available for use by each user. • Storage—Indicates whether a user is saved to the configuration file (nonvolatile) or not (volatile). Applies only to users with active status. • Status—Status of the conceptual row. Only rows with an active status are used by the SNMPv3 engine.
Group name	Name of the group to which this entry belongs.
Security model	Identifies the security model context for the security name.
Security name	Used with the security model; identifies a specific security name instance. Each security model/security name combination can be assigned to a specific group.
Storage type	Indicates whether a user is saved to the configuration file (nonvolatile) or not (volatile). Applies only to users with active status.
Status	Status of the conceptual row. Only rows with active status are used by the SNMPv3 engine.

Sample Output

show snmp v3

```

user@host> show snmp v3
Local engine ID: 80 00 0a 4c e04 31 32 33 34
Engine boots:      38
Engine time:       64583 seconds
Max msg size:      2048 bytes

Engine ID: local
  User          Auth/Priv  Storage  Status
  user1         md5/des   nonvolatile active
  user2         sha/none  nonvolatile active
  user3         none/none nonvolatile active

Engine ID: 81 00 0a 4c 04 64 64 64 64
  User          Auth/Priv  Storage  Status
  UNEW         md5/none  nonvolatile active
Group name      Security model  Security name      Storage type      Status
g1              usm          user1              nonvolatile active
g2              usm          user2              nonvolatile active
g3              usm          user3              nonvolatile active

Access control:
Group           Context prefix  Security model/level  Read view  Write view  Notify view
g1              usm/privacy  v1                   v1         v1
g2              usm/authent  v1                   v1         v1
g3              usm/none     v1                   v1         v1

```


PART 7

Index

- [Index on page 451](#)

Index

Symbols

#, comments in configuration statements.....	xxii
(), in syntax descriptions.....	xxii
/var/log/mib2d file.....	181
/var/log/snmpd file.....	181
< >, in syntax descriptions.....	xxii
[], in configuration statements.....	xxii
{ }, in configuration statements.....	xxii
(pipe), in syntax descriptions.....	xxii

A

AAA Objects MIB.....	34
Access Authentication Objects MIB.....	34
access statement	
usage guidelines.....	114
access-list statement.....	336
accounting options	
configuration.....	274
overview.....	269
accounting profiles	
filter.....	283
interface.....	281
MIB.....	293
Routing Engine.....	295
accounting-options statement.....	302
address statement	
SNMPv3.....	366
usage guidelines.....	125
address-mask statement.....	367
usage guidelines.....	125
agent, SNMP.....	12
agent-address statement.....	337
alarm management statement	
edit snmp.....	340
Alarm MIB.....	35
alarm statement	
RMON.....	324
usage guidelines.....	223
alarms, displaying	
health monitor.....	426
RMON.....	438

Analyzer MIB.....	35
Antivirus Objects MIB.....	35
archive-sites statement	
accounting.....	302
usage guidelines.....	280
ATM CoS MIB.....	35
ATM MIB.....	35
authentication-md5 statement.....	367
usage guidelines.....	111
authentication-none statement.....	368
usage guidelines.....	112
authentication-password statement.....	369
usage guidelines.....	111
authentication-sha statement.....	370
usage guidelines.....	111
authorization statement.....	342
usage guidelines.....	80

B

BFD MIB.....	35
BGP4 V2 MIB.....	35
braces, in configuration statements.....	xxii
brackets	
angle, in syntax descriptions.....	xxii
square, in configuration statements.....	xxii

C

categories statement.....	342
usage guidelines.....	92
Chassis Cluster MIB.....	36
Chassis Definitions for Router Model MIB.....	35
Chassis Forwarding MIB.....	36
Chassis MIB.....	36
Class 1 MIB objects.....	148
Class 2 MIB objects.....	152
Class 3 MIB objects.....	153
Class 4 MIB objects.....	154
Class-of-Service MIB.....	36
class-usage-profile statement.....	303
usage guidelines.....	291
clear	
snmp history.....	413
clear ilmi statistics command.....	412
clear snmp history command.....	413
clear snmp statistics command.....	414
client list	
adding to SNMP community.....	83
client-list statement.....	343
usage guidelines.....	83

client-list-name statement.....	343	dropped traffic	
usage guidelines.....	83	measuring.....	258
clients statement.....	344	Dynamic Flow Capture MIB.....	37
usage guidelines.....	80		
comments, in configuration statements.....	xxii		
commit-delay statement.....	344	E	
usage guidelines.....	79	engine-id statement	
community statement		SNMPv3.....	373
RMON.....	325	usage guidelines.....	108
usage guidelines.....	226	enterprise-oid statement.....	348
SNMP.....	345	Ethernet MAC MIB.....	37
usage guidelines.....	80	Event MIB.....	37
community string, SNMP.....	80	event statement.....	326
community-name statement.....	371	usage guidelines.....	226
usage guidelines.....	139	Experimental MIB.....	37
Configuration Management MIB.....	36		
contact statement.....	346	F	
usage guidelines.....	75, 77	falling-event-index statement.....	326
context (SNMPv3) statement	372	usage guidelines.....	223
conventions		falling-threshold statement	
text and syntax.....	xxi	health monitor.....	319
CoS		usage guidelines.....	265
measuring.....	255	RMON.....	327
MIB.....	36	falling-threshold-interval statement	
counters statement.....	304	RMON.....	328
curly braces, in configuration statements.....	xxii	usage guidelines.....	224
customer support.....	xxiii	fields statement	
contacting JTAC.....	xxiii	for interface profiles.....	305
		usage guidelines.....	281
		for Routing Engine profiles.....	306
		usage guidelines.....	296
D		file statement	
description statement		accounting (associating with profile).....	307
RMON.....	325	usage guidelines (filter profile).....	284
usage guidelines (alarms).....	223	usage guidelines (interface profile).....	281
usage guidelines (events).....	226	usage guidelines (MIB profile).....	294
SNMP.....	347	usage guidelines (Routing Engine	
usage guidelines.....	76, 77	profile).....	296
Destination Class Usage MIB.....	36	accounting (configuring log file).....	308
destination-classes statement.....	304	usage guidelines.....	278
usage guidelines.....	291	files statement.....	309
destination-port statement		filter profile.....	283
SNMP.....	348	filter-duplicates statement.....	349
usage guidelines.....	92	usage guidelines.....	79
DHCP MIB.....	36	filter-interfaces statement.....	350
DHCPv6 MIB.....	37	filter-profile statement.....	310
Digital Optical Monitoring MIB.....	37	usage guidelines.....	283
DNS Objects MIB.....	37	filtering get SNMP requests.....	79
documentation		Firewall MIB.....	37
comments on.....	xxiii	Flow Collection Services MIB.....	37

font conventions.....xxi

G

Get requests, SNMP.....9

group statement

 SNMPv3 (for access privileges)

 usage guidelines.....120

 SNMPv3 (for configuring).....374

 usage guidelines.....115

 SNMPv3 (for security name).....375

H

health metrics of network.....246

health monitor alarms, displaying.....426

health-monitor statement.....320

 usage guidelines.....265

Host Resources MIB.....38

I

IDP MIB.....38

ILMI.....6

 statistics

 clearing.....412

 displaying.....423

informs SNMP *See* SNMP informs

integrated local management interface *See* ILMI

Interface Accounting Forwarding Class MIB.....38

Interface MIB.....38

interface profile.....281

interface statement

 SNMP.....351

 usage guidelines.....95

interface-profile statement.....311

 usage guidelines.....281

interfaces limiting SNMP access.....95

interval statement

 accounting.....312

 usage guidelines (filter profile).....285

 usage guidelines (interface profile).....282

 usage guidelines (MIB profile).....294

 usage guidelines (Routing Engine profile).....296

 health monitor.....320

 usage guidelines.....265

 RMON.....328

 usage guidelines.....224

IP Forward MIB.....38

IPsec Generic Flow Monitoring Object MIB.....38

IPsec Monitoring MIB.....38

IPsec VPN Objects MIB.....39

IPv4 MIB.....39

IPv6 and ICMPv6 MIB.....39

IPv6 SNMP community string.....81

J

jnxRmonAlarmTable.....218

Juniper Networks MIB objects.....144

K

key performance indicators.....240

L

L2ALD MIB.....39

L2CP MIB.....39

L2TP MIB.....39

Layer 2 Control Protocol

 MIB.....39

LDP

 MIB.....39

License MIB.....40

local-engine statement.....377

location statement

 SNMP.....351

 usage guidelines.....76, 77

Logical Systems MIB.....40

logical-system statement.....352

logical-system-trap-filter statement.....353

LSYS MIB.....40

M

Management Information Base *See* MIBs

manuals

 comments on.....xxiii

master agent, SNMP.....12

measurement tests

 proxy ping.....244

message-processing-model statement.....378

 usage guidelines.....129

MIB object classes.....154

MIB profile.....293

mib-profile statement.....313

 usage guidelines.....293

MIBs

 AAA Objects.....34

 Access Authentication Objects.....34

 Alarm.....35

 Analyzer.....35

 Antivirus Objects.....35

ATM.....	35	NAT Objects.....	40
ATM CoS.....	35	NAT Resources-Monitoring.....	41
BFD.....	35	Optical Transport Network (OTN) Interface Management	41
BGP4 V2.....	35	OSPF.....	20
Chassis.....	36	Packet Forwarding Engine.....	41
Chassis Cluster.....	36	Packet Mirror.....	41
Chassis Definitions for Router Model.....	35	PAE Extension.....	41
Chassis Forwarding.....	36	Passive Monitoring.....	41
Class-of-Service.....	36	Ping.....	41
Configuration Management.....	36	use in ping test.....	164
Destination Class Usage.....	36	view configuration example, SNMP.....	98
DHCP	36	Policy Objects.....	41
DHCPv6	37	Power Supply Unit.....	42
Digital Optical Monitoring.....	37	PPP.....	19, 42
DNS Objects.....	37	PPPoE.....	42
Dynamic Flow Capture.....	37	Pseudowire ATM.....	42
Ethernet MAC.....	37	Pseudowire TDM.....	42
Event.....	37	QoS Interface.....	38
EX Series		Real-Time Performance Monitoring.....	42
Analyzer.....	35	Reverse-Path-Forwarding.....	42
PAE Extension.....	41	RMON Events and Alarms	43
Structure of Management Information	44	RPM.....	42
Virtual Chassis.....	45	RSVP	43
VLAN.....	45	Security Interface Extension Objects.....	43
Experimental.....	37	Security Screening Objects.....	43
Firewall.....	37	Services PIC.....	43
Flow Collection Services.....	37	SNMP IDP.....	38
Host Resources.....	38	SNMP object values, displaying.....	435
IDP.....	38	SONET APS.....	43
Interface.....	38	SONET/SDH Interface Management.....	44
IP Forward.....	38	Source Class Usage.....	44
IPsec Generic Flow Monitoring Object	38	SPU Monitoring.....	44
IPsec Monitoring.....	38	Structure of Management Information.....	44
IPsec VPN Objects.....	39	Junos OS for J Series and SRX Series devices, for.....	44
IPv4.....	39	Subscriber.....	44
IPv6 and ICMPv6.....	39	System Log.....	44
L2ALD.....	39	Traceroute.....	44
L2CP	39	Utility.....	45
L2TP.....	39	views	
Layer 2 Control Protocol.....	39	SNMP.....	97
LDP.....	39	Virtual Chassis.....	45
License.....	40	VLAN.....	45
Logical Systems.....	40	VPLS	45
LSYS.....	40	BGP MIB.....	45
MIMSTP.....	40	Generic MIB.....	45
MPLS.....	40	LDP MIB.....	45
MPLS LDP.....	40		
Multicast.....	24, 34		

VPN.....	45	operation statement.....	315
VPN Certificate Objects.....	45	for MIB profiles	
MIMSTP		usage guidelines.....	294
MIB.....	40	Optical Transport Network (OTN) Interface	
minimum accounting options configuration.....	276	Management MIB.	41
monitoring		OSPF MIB.....	20
service quality.....	239		
MPLS		P	
MIB.....	40	Packet Forwarding Engine MIB.....	41
MPLS LDP MIB.....	40	Packet Mirror MIB.....	41
Multicast MIB.....	24, 34	PAE Extension MIB.....	41
N		parameters statement.....	382
name statement.....	353	usage guidelines.....	128
usage guidelines.....	78	parentheses, in syntax descriptions.....	xxii
NAT Objects MIB.....	40	Passive Monitoring MIB.....	41
NAT Resources-Monitoring MIB.....	41	performance indicators.....	240
Network Address Translation Objects MIB See NAT		performance, monitoring.....	253
Objects MIB		Ping MIB.....	41
network health		use in ping test.....	164
measuring.....	246	view configuration example	
network performance		SNMP.....	98
measuring.....	253	pingCtlTable.....	244
nonpersistent statement.....	314	pingProbeHistoryTable.....	169
accounting		Policy Objects MIB.....	41
usage guidelines.....	278	port statement	
nonvolatile statement.....	354	SNMPv3.....	382
notify statement.....	379	usage guidelines.....	126
usage guidelines.....	122	Power Supply Unit MIB.....	42
notify-filter statement		PPP MIB.....	19, 42
for applying to target.....	380	PPPoE MIB.....	42
usage guidelines.....	129	prefix list	
for configuring.....	380	adding to SNMP community.....	83
usage guidelines.....	124	privacy-3des statement.....	383
notify-view statement.....	381	usage guidelines.....	113
usage guidelines.....	117	privacy-aes128 statement.....	384
O		usage guidelines.....	112
object-names statement.....	314	privacy-des statement.....	385
objects-names statement		usage guidelines.....	113
for Routing Engine profiles		privacy-none statement.....	385
usage guidelines.....	295	usage guidelines.....	113
oid statement		privacy-password statement.....	386
SNMP.....	354	usage guidelines	
usage guidelines.....	97	for 3DES algorithm.....	113
SNMPv3.....	381	for AES algorithm.....	112
usage guidelines.....	124	for DES algorithm.....	113
		profiles, accounting	
		filter.....	283
		interface.....	281

MIB.....	293
Routing Engine.....	295
proxy ping	
measurement tests.....	244
Pseudowire ATM MIB.....	42
Pseudowire TDM MIB.....	42
PSU MIB.....	42

R

read-view statement.....	387
usage guidelines.....	117
real-time performance monitoring	
in service provider networks.....	244
Real-Time Performance Monitoring MIB.....	42
Remote Monitoring.....	438
remote operations MIBs.....	163
remote-engine statement.....	388
request snmp spoof-trap command.....	416
request-type statement.....	329
RMON	
usage guidelines.....	225
retry-count statement.....	375
usage guidelines.....	136
Reverse-Path-Forwarding MIB.....	42
rising-event-index statement.....	330
usage guidelines.....	223
rising-threshold statement	
health monitor.....	321
RMON.....	330
RMON alarm entries.....	222
RMON alarms.....	217, 237
RMON alarms and events, displaying.....	438
RMON event entries.....	226
RMON events.....	219, 236
RMON Events and Alarms MIB.....	43
rmon statement.....	331
usage guidelines.....	236
Routing Engine profile.....	295
routing instances	
access lists	
configuring.....	159
SNMP	
enabling access.....	157
identifying.....	156
specifying.....	157
routing-engine-profile statement.....	315
usage guidelines.....	295

routing-instance statement	
SNMP.....	356
SNMPv3.....	389
usage guidelines.....	126
routing-instance-access.....	357
RPM MIB.....	42
RSVP MIB.....	43

S

sample-type statement.....	331
usage guidelines	
for alarms.....	225
for events.....	226
Security Interface Extension Objects MIB.....	43
Security Screening Objects MIB.....	43
security-level statement	
for access privileges.....	390
usage guidelines.....	116
for SNMP notifications.....	391
usage guidelines.....	130
security-model statement	
for access privileges.....	392
usage guidelines.....	116
for groups.....	393
usage guidelines.....	119
for SNMP notifications.....	394
usage guidelines.....	130
security-name statement	
for community string.....	395
for security group.....	396
usage guidelines.....	120
for SNMP notifications.....	397
usage guidelines.....	130
security-to-group statement.....	398
usage guidelines.....	114
service quality	
monitoring.....	239
Services PIC MIB.....	43
Set requests, SNMP.....	9
show ilmi statistics command.....	423
show interfaces snmp-index command.....	422
show snmp health-monitor command.....	426
show snmp inform-statistics command.....	433
show snmp mib command.....	435
show snmp rmon command.....	438
show snmp statistics command.....	442
show snmp v3 command.....	446

size statement		snmp statement.....	357
accounting.....	316	usage guidelines	
usage guidelines.....	279	SNMPv1 and SNMPv2.....	73
SNMP.....	86	SNMPv3.....	102
adding client lists and prefix lists.....	83	SNMP traps.....	10
agent.....	9, 12	source address configuration.....	89
architecture.....	9	spoofing.....	416
commit delay timer.....	79	system logging severity levels.....	11
community string.....	80	snmp-community statement.....	399
configuration		SNMPv2	
version 3.....	102	Passive Monitoring Traps MIB.....	92
versions 1 and 2.....	73	SNMPv3	
FAQs		authentication, configuring.....	111
troubleshooting.....	187	informs, configuring.....	131
filtering duplicate requests.....	79	local engine ID, configuring.....	108
health monitor alarms, displaying.....	426	minimum configuration.....	104
inform statistics, displaying.....	433	SNMPv3 context	
limiting interface access.....	95	usage guidelines.....	139
logging, enabling.....	164	SONET APS MIB.....	43
manager.....	9	SONET Automatic Protection Switching MIB.....	43
master agent.....	12	SONET/SDH Interface Management MIB.....	44
MIB object values, displaying.....	435	Source Class Usage MIB.....	44
MIB views.....	97	source-address statement.....	358
remote operations.....	161	usage guidelines.....	89
RMON alarms and events, displaying.....	438	source-classes statement.....	316
standards documents.....	15	usage guidelines.....	291
statistics		SPU Monitoring MIB.....	44
clearing.....	414	standards documents	
displaying.....	442	SNMP and MIBs.....	16
subagent.....	12	start-time statement	
system contact.....	75, 77	accounting.....	317
system description.....	76, 77	usage guidelines.....	279
system location.....	76, 77, 351	startup-alarm statement.....	332
system name.....	78	usage guidelines.....	225
tracing operations.....	181	Structure of Management Information MIB.....	44
trap groups.....	92	for EX Series.....	44
trap notification for remote operations.....	163	Junos OS for J Series and SRX Series devices,	
trap options.....	88	for.....	44
version 3 configuration, displaying.....	446	subagent, SNMP.....	12
views, setting.....	162	Subscriber MIB.....	44
SNMP FAQs		support, technical See technical support	
troubleshooting		syntax conventions.....	xxi
best practices.....	188	sysContact object, MIB II.....	75, 77
snmp history		sysDescription object, MIB II.....	76, 77
clear.....	413	sysLocation object, MIB II.....	76, 77
SNMP index		syslog-subtag statement.....	332
interface information, displaying.....	422	usage guidelines.....	226
SNMP inform statistics, displaying.....	433	sysName object, MIB II.....	78
SNMP informs.....	131	system contact, SNMP.....	75, 77

system description, SNMP.....	76, 77
system location, SNMP.....	76, 77, 351
System Log MIB.....	44
system logging severity levels, SNMP traps.....	11
system name, SNMP.....	78

T

tag statement.....	399
SNMPv3	
usage guidelines.....	140
usage guidelines.....	122
tag-list statement.....	400
usage guidelines.....	126
target-address statement.....	401
usage guidelines.....	124
target-parameters statement.....	402
usage guidelines.....	128
targets statement.....	358
usage guidelines.....	92
technical support	
contacting JTAC.....	xxiii
timeout statement.....	376
usage guidelines.....	136
traceoptions statement.....	359
SNMP	
usage guidelines.....	181
Traceroute MIB.....	44, 171
traceRouteHopsTable.....	176
tracing operations	
SNMP.....	181
transfer-interval statement	
accounting.....	317
usage guidelines.....	279
trap groups, SNMP.....	92
trap notification for SNMP remote operations.....	163
trap-group statement.....	361
usage guidelines.....	92
trap-groups statement	
usage guidelines.....	86
trap-options statement.....	362
usage guidelines.....	86, 88
traps	
definition.....	10
spoofing.....	416
type statement.....	333
usage guidelines.....	122

U

user statement	
SNMPv3.....	403
usm statement.....	404
Utility MIB.....	45

V

v3 statement.....	406
usage guidelines.....	102
vacm statement.....	408
usage guidelines.....	114
var/log/mib2d file.....	181
var/log/snmpd file.....	181
variable statement.....	333
usage guidelines.....	226
variable-length string indexes.....	163
version statement	
SNMP.....	363
usage guidelines.....	92
view statement	
SNMP (associating with community).....	363
usage guidelines.....	80
SNMP (configuring MIB view).....	364
usage guidelines.....	97
views, MIB	
SNMP.....	97, 162
Virtual Chassis MIB.....	45
VLAN MIB.....	45
VPLS MIBs.....	45
VPN Certificate Objects MIB.....	45
VPN MIB.....	45

W

write-view statement.....	409
usage guidelines.....	118