Junos® OS

MIB Reference for SRX1400, SRX3400, and SRX3600 Services Gateways

Release
12.1

Published: 2012-03-06

Copyright © 2012, Juniper Networks, Inc.

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

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

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

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

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

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

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

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

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

MIB Reference for SRX1400, SRX3400, and SRX3600 Services Gateways
Release 12.1
Copyright © 2012, Juniper Networks, Inc.
All rights reserved.

Revision History
March 2012— R1 Junos OS 12.1

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 This Guide . . . xvii

SRX Series Documentation and Release Notes . . . xvii
Objectives . . . xvii
Audience . . . xviii
Documentation Conventions . . . xviii
Requesting Technical Support . . . xx
Self-Help Online Tools and Resources . . . xx
Opening a Case with JTAC . . . xx

Part 1 Network Management Introduction

Chapter 1 **Network Management Overview . . . 3**

Understanding Device Management Functions in Junos OS . . . 3

Chapter 2 **Network Management Configuration Statements . . . 5**

Configuration Statements at the [edit accounting-options] Hierarchy Level . . . 5
Configuration Statements at the [edit snmp] Hierarchy Level . . . 6

Part 2 Simple Network Management Protocol (SNMP)

Chapter 3 **SNMP Overview . . . 13**

Understanding the SNMP Implementation in Junos OS . . . 13
SNMP Architecture . . . 13
SNMP MIBs . . . 14
SNMP Traps and Informs . . . 14
Junos OS SNMP Agent Features . . . 16
Standard SNMP MIBs Supported by Junos OS . . . 17
Juniper Networks Enterprise-Specific MIBs . . . 31
Loading MIB Files to a Network Management System . . . 43

Chapter 4 **Configuring SNMP . . . 47**

Configuring SNMP on a Device Running Junos OS . . . 48
Configuring the System Contact on a Device Running Junos OS . . . 50
Configuring the System Location for a Device Running Junos OS . . . 50
Configuring the System Description on a Device Running Junos OS . . . 51
Filtering Duplicate SNMP Requests . . . 51
Configuring the Commit Delay Timer . . . 52
Configuring the System Name . . . 52
Examples: Configuring the SNMP Community String . . . 53
Example: Configuring SNMP Trap Groups . . . 54
Configuring the SNMP Community String . . . 54

Adding a Group of Clients to an SNMP Community 55
Configuring SNMP Trap Options and Groups on a Device Running Junos OS 56
Configuring SNMP Trap Options 57
Configuring the Source Address for SNMP Traps 58
Configuring the Agent Address for SNMP Traps 59
Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps 60
Configuring SNMP Trap Groups 60
Spoofing Enterprise-Specific SNMP Traps 63
Configuring the Interfaces on Which SNMP Requests Can Be Accepted 63
Example: Configuring Secured Access List Checking 64
Example: Ping Proxy MIB 64
Filtering Interface Information Out of SNMP Get and GetNext Output 65
Example: Tracing SNMP Activity 65
Configuring MIB Views 66
Tracing SNMP Activity on a Device Running Junos OS 67
Configuring the Number and Size of SNMP Log Files 68
Configuring Access to the Log File 68
Configuring a Regular Expression for Lines to Be Logged 68
Configuring the Trace Operations 69
Configuring the Local Engine ID 70

Chapter 5 SNMPv3 Overview 73

SNMPv3 Overview 73

Chapter 6 Configuring SNMPv3 75

Complete SNMPv3 Configuration Statements 76
Minimum SNMPv3 Configuration on a Device Running Junos OS 78
Configuring the Local Engine ID 79
Creating SNMPv3 Users 80
Configuring the SNMPv3 Authentication Type 81
Configuring MD5 Authentication 81
Configuring SHA Authentication 81
Configuring No Authentication 82
Configuring the Encryption Type 82
Configuring the Advanced Encryption Standard Algorithm 83
Configuring the Data Encryption Algorithm 83
Configuring Triple DES 83
Configuring No Encryption 84
Defining Access Privileges for an SNMP Group 84
Configuring the Access Privileges Granted to a Group 85
Configuring the Group 86
Configuring the Security Model 86
Configuring the Security Level 86
Associating MIB Views with an SNMP User Group 87
Configuring the Notify View 87
Configuring the Read View 88
Configuring the Write View 88
Example: Access Privilege Configuration 88

 Copyright © 2012, Juniper Networks, Inc.

Assigning Security Model and Security Name to a Group 89
Configuring the Security Model 90
Assigning Security Names to Groups 90
Configuring the Group 90
Example: Security Group Configuration 91
Configuring SNMPv3 Traps on a Device Running Junos OS 91
Configuring the SNMPv3 Trap Notification 93
Example: Configuring SNMPv3 Trap Notification 93
Configuring the Trap Notification Filter 94
Configuring the Trap Target Address 95
Configuring the Address 96
Configuring the Address Mask 96
Configuring the Port 96
Configuring the Routing Instance 96
Configuring the Trap Target Address 96
Applying Target Parameters 97
Example: Configuring the Tag List 98
Defining and Configuring the Trap Target Parameters 98
Applying the Trap Notification Filter 99
Configuring the Target Parameters 99
Configuring the Message Processing Model 100
Configuring the Security Model 100
Configuring the Security Level 100
Configuring the Security Name 101
Configuring SNMP Informs 101
Configuring the Remote Engine and Remote User 102
Example: Configuring the Remote Engine ID and Remote Users 103
Configuring the Inform Notification Type and Target Address 104
Example: Configuring the Inform Notification Type and Target Address 105
Configuring the SNMPv3 Community 106
Configuring the Community Name 106
Configuring the Security Names 107
Configuring the Tag 107
Example: SNMPv3 Community Configuration 108
Example: SNMPv3 Configuration 108

Chapter 7 Standard SNMP Traps 113

Standard SNMP Traps Supported on Devices Running Junos OS 113
Standard SNMP Version 1 Traps 113
Standard SNMP Version 2 Traps 117
SNMP Version 2 MPLS Traps 120
Unsupported Standard SNMP Traps 121

Chapter 8 Juniper Networks Enterprise-Specific SNMP Traps 125

Juniper Networks Enterprise-Specific SNMP Traps 125
Juniper Networks Enterprise-Specific SNMP Version 1 Traps 126
Juniper Networks Enterprise-Specific SNMP Version 2 Traps 133
Juniper Networks Enterprise-Specific BGP Traps 140
Juniper Networks Enterprise-Specific LDP Traps 141
Disabling LDP Traps 141

Copyright © 2012, Juniper Networks, Inc.

Juniper Networks Enterprise-Specific Traps on EX Series Switches 141
Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers 142
Standard SNMP Traps on EX Series Ethernet Switches 142
Configuring SNMPv3 Traps on a Device Running Junos OS 143

Chapter 9 **Summary of SNMP Configuration Statements 145**

access-list 145
agent-address 146
authorization 146
categories 147
client-list 147
client-list-name 148
clients 148
commit-delay 149
community 150
contact 151
description 151
destination-port 152
engine-id 152
filter-duplicates 152
filter-interfaces 153
interface 154
location 154
logical-system 155
logical-system-trap-filter 156
name 156
nonvolatile 157
oid 157
routing-instance 158
routing-instance-access 159
snmp 159
source-address 160
targets 161
traceoptions 162
trap-group 164
trap-options 165
version 166
view 167
view (Associating a MIB View with a Community) 167
view (Configuring a MIB View) 168

Chapter 10 **SNMP Remote Operations 169**

SNMP Remote Operations Overview 169
SNMP Remote Operation Requirements 170
Setting SNMP Views 170
Example: Setting SNMP Views 170
Setting Trap Notification for Remote Operations 170
Example: Setting Trap Notification for Remote Operations 171

 Copyright © 2012, Juniper Networks, Inc.

Using Variable-Length String Indexes 171
Example: Set Variable-Length String Indexes 171
Enabling Logging 172
Using the Ping MIB for Remote Monitoring Devices Running Junos OS 172
Starting a Ping Test 172
Using Multiple Set Protocol Data Units (PDUs) 173
Using a Single Set PDU 173
Monitoring a Running Ping Test 174
pingResultsTable 174
pingProbeHistoryTable 175
Generating Traps 176
Gathering Ping Test Results 177
Stopping a Ping Test 178
Interpreting Ping Variables 178

Chapter 11 SNMP Support for Routing Instances 181

Understanding SNMP Support for Routing Instances 181
Identifying a Routing Instance 182
Enabling SNMP Access over Routing Instances 183
Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community 184
Example: Configuring Interface Settings for a Routing Instance 185
Configuring Access Lists for SNMP Access over Routing Instances 186
Trap Support for Routing Instances 187
MIB Support Details 187

Part 3 SRX1400, SRX3400 and SRX3600 Services Gateways MIBs

Chapter 12 SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs 201

List of SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs 201

Chapter 13 Interpreting the Structure of Management Information MIB 207

Structure of Management Information MIB 207
jnxProducts 208
jnxServices 208
jnxMibs 209
jnxTraps 211
jnxExperiment 212

Chapter 14 Interpreting the Enterprise-Specific AAA Objects MIB 213

AAA Objects MIB 213
AAA Objects MIB Textual Conventions 214
jnxUserAAAStatTable 216
jnxUserAAAServerName 216
Access Authentication-Related Traps 217

Copyright © 2012, Juniper Networks, Inc.

Chapter 15 Interpreting the Enterprise-Specific Access Authentication Objects MIB 219
Access Authentication Objects MIB 219
jnxJsFwAuthStats 219
jnxJsAuthTrapVars 220
jnxJsAuthNotifications 221
Chapter 16 Interpreting the Enterprise-Specific Alarm MIB 223
Alarm MIB 223
jnxAlarmRelayMode 223
jnxYellowAlarms 224
jnxRedAlarms 225
Chapter 17 Interpreting the Enterprise-Specific ATM Class-of-Service MIB 227
ATM Class-of-Service MIB 227
jnxCosAtmVcTable 228
jnxCosAtmVcScTable 228
jnxCosAtmVcQstatsTable 230
jnxCosAtmTrunkTable 231
Chapter 18 Interpreting the Enterprise-Specific BGP4 V2 MIB 235
BGP4 V2 MIB 235
jnxBgpM2PrefixCountersTable 236
Chapter 19 Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB 237
Bidirectional Forwarding Detection MIB 237
jnxBfdSessTable 237
Notifications for the BFD MIB 238
Chapter 20 Interpreting the Enterprise-Specific Chassis MIBs 239
Chassis MIBs 239
Chassis MIB Textual Conventions 240
jnxBoxAnatomy 241
Top-Level Objects 241
jnxContainersTable 242
jnxContentsLastChange 249
jnxContentsTable 249
jnxLEDLastChange 267
jnxLEDTable 267
jnxFilledLastChange 270
jnxFilledTable 271
jnxOperatingTable 287
jnxRedundancyTable 299
jnxFruTable 306
jnxBoxKernelMemoryUsedPercent 355
jnxBoxSystemDomainType 355
Chassis Traps 355
SNMPv1 Trap Format 358
SNMPv2 Trap Format 359
Chassis Definitions for Router Model MIB 360

 Copyright © 2012, Juniper Networks, Inc.

MIB Objects for the SRX1400 Services Gateway 361
MIB Objects for the SRX3400 Services Gateway 362
MIB Objects for the SRX3600 Services Gateway 363

Chapter 21 Interpreting the Enterprise-Specific Chassis Cluster MIBs 365

Chassis Cluster MIB 365
jnxJsChassisClusterSwitchover 365
jnxJsChClusterIntfTrap 366
Chassis Cluster Notifications 367

Chapter 22 Interpreting the Enterprise-Specific Configuration Management MIB . . . 369

Configuration Management MIB 369
Configuration Management MIB Textual Conventions 369
Configuration Change Management Objects and jnxCmCfgChgEventTable . . . 370
jnxCmCfgChgEventTable 371
Rescue Configuration Change Management Objects 372
Configuration Management Notifications 373

Chapter 23 Interpreting the Enterprise-Specific Destination Class Usage MIB 375

Destination Class Usage MIB 375
jnxDCUsTable 376
jnxDcuStatsTable 376

Chapter 24 Interpreting the Enterprise-Specific DNS Objects MIB 379

DNS Objects MIB 379
jnxJsDnsProxyDataObjects 379

Chapter 25 Interpreting the Enterprise-Specific Ethernet MAC MIB 381

Ethernet MAC MIB 381
jnxMacStatsTable 381

Chapter 26 Interpreting the Enterprise-Specific Event MIB 383

Event MIB 383
jnxEventAvTable 383
Notifications for the Event MIB 384

Chapter 27 Interpreting the Enterprise-Specific Firewall MIB 385

Firewall MIB 385
jnxFirewallsTable 386
jnxFirewallCounterTable 386

Chapter 28 Interpreting the Enterprise-Specific Host Resources MIB 389

Host Resources MIB 389
jnxHrStorageTable 389

Chapter 29 Interpreting the Enterprise-Specific Interface MIB 391

Interface MIB 391
jnxIfTable 391
ifChassisTable 394

Chapter 30 Interpreting the Enterprise-Specific IP Forward MIB 397

IP Forward MIB 397
jnxIpCidrRouteTable 398

Copyright © 2012, Juniper Networks, Inc.

Chapter 31 Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object MIB 399
IPsec Generic Flow Monitoring Object MIB 399
Branch Tree Objects 400
IPSec Generic Flow Monitoring Object MIB Textual Conventions 400
Number of IKE Tunnels Currently Active 404
IPsec Phase 1 IKE Tunnel Table 404
IPsec Phase 2 IKE Tunnel Table 408
IPsec Phase 2 Security Association Table 411
Chapter 32 Interpreting the Enterprise-Specific IPsec Monitoring MIB 415
IPSec Monitoring MIB 415
jnxIkeTunnelTable 415
jnxIPSecTunnelTable 418
jnxIPSecSaTable 421
Chapter 33 Interpreting the Enterprise-Specific IPv4 MIB 423
IPv4 MIB 423
jnxIpv4AddrTable 423
Chapter 34 Interpreting the Enterprise-Specific IPv6 MIB 425
IPv6 MIB 425
jnxIpv6GlobalStats 425
jnxIcmpv6GlobalStats 428
Chapter 35 Interpreting the Enterprise-Specific License MIB 433
License MIB 433
jnxLicenseInstallTable 434
Supported Features and Configuration Parameters for Licensing 434
jnxLicenseFeatureListTable 434
License Configuration Parameters 435
Juniper Networks Enterprise-Specific License MIB Notifications 435
Chapter 36 Interpreting the Enterprise-Specific Logical Systems (LSYS) MIBs 437
LSYS Security Profile 437
jnxLsysSecurityProfile 437
LSYS Zone MIB 438
jnxLsysSpZoneMIB 438
jnxLsysSpZoneObjects 439
jnxLsysSpZoneSummary 440
LSYS Scheduler MIB 440
jnxLsysSpSchedulerMIB 441
jnxLsysSpSchedulerObjects 441
jnxLsysSpSchedulerSummary 442
LSYS Policy MIB 443
jnxLsysSpPolicyMIB 443
jnxLsysSpPolicyObjects 444
jnxLsysSpPolicySummary 444

 Copyright © 2012, Juniper Networks, Inc.

LSYS Policy with Count MIB 445
jnxLsysSpPolicywcntMIB 445
jnxLsysSpPolicywcntObjects 446
jnxLsysSpPolicywcntSummary 447
LSYS Flow Gate MIB 448
jnxLsysSpFlowgateMIB 448
jnxLsysSpFlowgateObjects 448
jnxLsysSpFlowgateSummary 449
LSYS Flow Session MIB 450
jnxLsysSpFlowsessMIB 450
jnxLsysSpFlowsessObjects 451
jnxLsysSpFlowsessSummary 451
LSYS AUTH Entry MIB 452
jnxLsysSpAuthentryMIB 452
jnxLsysSpAuthentryObjects 453
jnxLsysSpAuthentrySummary 453
LSYS NAT Source Pool MIB 454
jnxLsysSpNATsrcpoolMIB 454
jnxLsysSpNATsrcpoolObjects 455
jnxLsysSpNATsrcpoolSummary 456
LSYS NAT Destination Pool MIB 456
jnxLsysSpNATdstpoolMIB 457
jnxLsysSpNATdstpoolObjects 457
jnxLsysSpNATdstpoolSummary 458
LSYS NAT Source PAT Address MIB 459
jnxLsysSpNATsrcpatadMIB 459
jnxLsysSpNATsrcpatadObjects 459
jnxLsysSpNATsrcpatadSummary 460
LSYS NAT Source No PAT Address MIB 461
jnxLsysSpNATsrcnopatadMIB 461
jnxLsysSpNATsrcnopatadObjects 462
jnxLsysSpNATsrcnopatadSummary 462
LSYS NAT Source Rule MIB 463
jnxLsysSpNATsrcruleMIB 463
jnxLsysSpNATsrcruleObjects 464
jnxLsysSpNATsrcruleSummary 465
LSYS NAT Destination Rule MIB 466
jnxLsysSpNATdstruleMIB 466
jnxLsysSpNATdstruleObjects 467
jnxLsysSpNATdstruleSummary 467
LSYS NAT Static Rule MIB 468
jnxLsysSpNATstaticruleMIB 468
jnxLsysSpNATstaticruleObjects 469
jnxLsysSpNATstaticruleSummary 470
LSYS NAT Cone Bind MIB 471
jnxLsysSpNATconebindMIB 471
jnxLsysSpNATconebindObjects 471
jnxLsysSpNATconebindSummary 472

Copyright © 2012, Juniper Networks, Inc.

LSYS NAT Port Overloading IP Number MIB 473
jnxLsysSpNATpoipnumMIB 473
jnxLsysSpNATpoipnumObjects 474
jnxLsysSpNATpoipnumSummary 474
LSYS CPU MIB 475
jnxLsysSpCPUMIB 475
jnxLsysSpCPUObjects 476
jnxLsysSpCPSummary 477
jnxLsysSpSPUSummary 477

Chapter 37 Interpreting the Enterprise-Specific Network Address Translation Objects MIB 479

NAT Objects MIB 479
Source NAT Table 480
New Source NAT Table 481
jnxJsNatIfSrcPoolPortTable 483
NAT Rule Hit Table 484
NAT Pool Hit Table 484
NAT Trap Definitions 485

Chapter 38 Interpreting the Enterprise-Specific Packet Forwarding Engine MIB 487

Packet Forwarding Engine MIB 487
jnxPfeNotifyGlTable 487
jnxPfeNotifyTypeTable 489

Chapter 39 Interpreting the Enterprise-Specific Ping MIB 491

PING MIB 491
jnxPingCtlTable 492
jnxPingResultsTable 496
jnxPingProbeHistoryTable 500
jnxPingLastTestResultTable 502

Chapter 40 Interpreting the Enterprise-Specific Policy Objects MIB 507

Policy Objects MIB 507
Security Policy Table 508
jnxJsPolicyStatsTable 509

Chapter 41 Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB 513

Reverse Path Forwarding MIB 513
jnxRpfStatsTable 513

Chapter 42 Interpreting the Enterprise-Specific RMON Events and Alarms MIB 515

RMON Events and Alarms MIB 515
jnxRmonAlarmTable 516
RMON Event and Alarm Traps 517

Chapter 43 Interpreting the Enterprise-Specific Security Interface Extension Objects MIB 519

Security Interface Extension Objects MIB 519
jnxJsIfMonTable 519

 Copyright © 2012, Juniper Networks, Inc.

Chapter 44 Interpreting the Enterprise-Specific Security Screening Objects MIB . . . 523
Security Screening Objects MIB . . . 523
jnxJsScreenMonTable . . . 524
jnxJsScreenSweepTable . . . 534
jnxJsScreenNotifications . . . 535
jnxJsScreenTrapVars . . . 535
Chapter 45 Interpreting the Enterprise-Specific SNMP IDP MIB . . . 537
SNMP IDP MIB . . . 537
jnxJsIdpAttackTable . . . 537
IDP Signature Update and Attack Traps . . . 538
Chapter 46 Interpreting the Enterprise-Specific Source Class Usage MIB . . . 539
Source Class Usage MIB . . . 539
jnxScuStatsTable . . . 539
Chapter 47 Interpreting the Enterprise-Specific SPU Monitoring MIB . . . 541
SPU Monitoring Objects MIB . . . 541
SPU Monitoring Objects Table . . . 542
Cluster Objects Table . . . 543
Chapter 48 Interpreting the Enterprise-Specific System Log MIB . . . 545
System Log MIB . . . 545
jnxSyslogTable . . . 546
jnxSyslogAvTable . . . 548
Chapter 49 Interpreting the Enterprise-Specific Traceroute MIB . . . 549
Traceroute MIB . . . 549
jnxTraceRouteCtlTable . . . 549
Chapter 50 Interpreting the Enterprise-Specific Utility MIB . . . 551
Utility MIB . . . 551
jnxUtilCounter32Table . . . 552
jnxUtilCounter64Table . . . 552
jnxUtilIntegerTable . . . 553
jnxUtilUintTable . . . 553
jnxUtilStringTable . . . 554
Chapter 51 Interpreting the VPN Certificate Objects MIB . . . 555
VPN Certificate Objects MIB . . . 555
jnxJsLoadedCaCertTable . . . 555
jnxJsLoadedLocalCertTable . . . 556

Part 4 Accounting Options

Chapter 52 Accounting Options Overview . . . 561
Accounting Options Overview . . . 561

Chapter 53 Configuring Accounting Options 563
Accounting Options Configuration 563
Accounting Options—Full Configuration 563
Minimum Accounting Options Configuration 564
Configuring Accounting-Data Log Files 566
Configuring the Storage Location of the File 567
Configuring the Maximum Size of the File 567
Configuring the Maximum Number of Files 568
Configuring the Start Time for File Transfer 568
Configuring the Transfer Interval of the File 568
Configuring Archive Sites 569
Configuring the Interface Profile 569
Configuring Fields 570
Configuring the File Information 570
Configuring the Interval 570
Example: Configuring the Interface Profile 571
Configuring the Filter Profile 572
Configuring the Counters 572
Configuring the File Information 572
Configuring the Interval 573
Example: Configuring a Filter Profile 574
Example: Configuring Interface-Specific Firewall Counters and Filter Profiles . . 574
Understanding Source Class Usage and Destination Class Usage Options 576
Configuring SCU or DCU 577
Creating Prefix Route Filters in a Policy Statement 577
Applying the Policy to the Forwarding Table 577
Enabling Accounting on Inbound and Outbound Interfaces 578
Configuring SCU on a Virtual Loopback Tunnel Interface 579
Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC 579
Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface 579
Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface 580
Configuring Class Usage Profiles 580
Configuring a Class Usage Profile 581
Configuring the File Information 581
Configuring the Interval 581
Creating a Class Usage Profile to Collect Source Class Usage Statistics . . . 581
Creating a Class Usage Profile to Collect Destination Class Usage Statistics 582
Configuring the MIB Profile 583
Configuring the File Information 583
Configuring the Interval 583
Configuring the MIB Operation 584
Configuring MIB Object Names 584
Example: Configuring a MIB Profile 584

 Copyright © 2012, Juniper Networks, Inc.

Configuring the Routing Engine Profile 584
Configuring Fields 585
Configuring the File Information 585
Configuring the Interval 585
Example: Configuring a Routing Engine Profile 586

Part 5 Index

Index 589

Copyright © 2012, Juniper Networks, Inc.

About This Guide

This preface provides the following guidelines for using the *MIB Reference for SRX1400, SRX3400, and SRX3600 Services Gateways*:

- SRX Series Documentation and Release Notes on page xvii
- Objectives on page xvii
- Audience on page xviii
- Documentation Conventions on page xviii
- Requesting Technical Support on page xx

SRX Series Documentation and Release Notes

For a list of related SRX Series documentation, see http://www.juniper.net/techpubs/hardware/srx-series-main.html .

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

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

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

Objectives

This guide provides an overview of the SRX1400, SRX3400, and SRX3600 Services Gateways MIB specifications of the Simple Network Management Protocol (SNMP). The SRX1400, SRX3400, and SRX3600 Services Gateways information is stored in the form of a virtual storage area called a Management Information Base (MIB). Each MIB contains several MIB objects that describe the services gateway components.

Copyright © 2012, Juniper Networks, Inc.

NOTE: For information about managing networks with Junos OS, see *Junos Network Management Configuration Guide*. For information about the SNMP MIBs and Traps supported by Junos OS, see *Junos SNMP MIBs and Traps Reference*.

Audience

This guide is designed for network administrators to configure and monitor the SRX1400, SRX3400, and SRX3600 Services Gateways. This guide may also be useful for application developers who are developing management applications for SRX1400, SRX3400, and SRX3600 Services Gateways.

Documentation Conventions

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

Table 1: Notice Icons

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

Table 2 on page xviii 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`

Copyright © 2012, Juniper Networks, Inc.

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

Convention	Description	Examples
Italic text like this	• Introduces important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub** <**default-metric** *metric*>;
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** (*string1* | *string2* | *string3*)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** *community-ids* **]**

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

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

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: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

Copyright © 2012, Juniper Networks, Inc.

PART 1

Network Management Introduction

- Network Management Overview on page 3
- Network Management Configuration Statements on page 5

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 1

Network Management Overview

- Understanding Device Management Functions in Junos OS on page 3

Understanding Device Management Functions in Junos OS

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

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

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

Table 3: Device Management Features in Junos OS

Task	Junos OS Feature
Fault management	Monitor and see faults using: • Operational mode commands—For more information about operational mode commands, see the *CLI User Guide*. • SNMP MIBs—For more information about SNMP MIBs supported by Junos OS, see "Standard SNMP MIBs Supported by Junos OS" on page 17 and "Juniper Networks Enterprise-Specific MIBs" on page 31. • Standard SNMP traps—For more information about standard SNMP traps, see the "Standard SNMP Traps Supported on Devices Running Junos OS" on page 113. • Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 125. • System log messages—For more information about how to configure system log messages, see the *Junos OS System Basics Configuration Guide*. For more information about how to view system log messages, see the *Junos OS System Log Messages Reference*.

Copyright © 2012, Juniper Networks, Inc.

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

Task	Junos OS Feature
Configuration management	• Configure device attributes using the command-line interface (CLI). For more information about configuring the device using the CLI, see the *CLI User Guide*. • Configuration Management MIB—For more information about the Configuration Management MIB, see the "Configuration Management MIB" on page 369.
Accounting management	Perform the following accounting-related tasks: • Collect statistics for interfaces, firewall filters, destination classes, source classes, and the Routing Engine. For more information about collecting statistics, see "Accounting Options Configuration" on page 563. • 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. • Count packets as part of a firewall filter. For more information about firewall filter policies, see "Juniper Networks Enterprise-Specific MIBs" on page 31 and the *Routing Protocols and Policies Configuration Guide for Security Devices*.
Performance management	Monitor performance in the following ways: • Use operational mode commands. For more information about monitoring performance using operational mode commands, see the *CLI User Guide*. • Use firewall filters. For more information about performance monitoring using firewall filters, see the *Routing Protocols and Policies Configuration Guide for Security Devices*.
Security management	Assure security in your network in the following ways: • Control access to the router and authenticate users. For more information about access control and user authentication, see the *Junos OS Security Configuration Guide*. • Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see "Configuring the Local Engine ID" on page 70 and "Tracing SNMP Activity on a Device Running Junos OS" on page 67.

Related Documentation

- Understanding the SNMP Implementation in Junos OS on page 13
- Accounting Options Overview on page 561

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 2

Network Management Configuration Statements

- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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

This topic shows all possible configuration statements at the **[edit accounting-options]** hierarchy level and their level in the configuration hierarchy. When you are configuring Junos OS, your current hierarchy level is shown in the banner on the line preceding the **user@host#** prompt.

For a list of the complete configuration statement hierarchy, see the *Junos OS Configuration Statements and Commands*.

```
[edit]
accounting-options {
   class-usage-profile profile-name {
      file filename;
      interval minutes;
      destination-classes {
         destination-class-name;
      }
      source-classes {
         source-class-name;
      }
   }
   file filename {
      archive-sites {
      }
      files number;
      nonpersistent;
      size bytes;
      start-time time;
      transfer-interval minutes;
   }
   filter-profile profile-name {
      counters {
         counter-name;
      }
```

Copyright © 2012, Juniper Networks, Inc.

```
        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 561
- Accounting Options Configuration on page 563

Configuration Statements at the [edit snmp] Hierarchy Level

This topic shows all possible configuration statements at the **[edit snmp]** hierarchy level and their level in the configuration hierarchy. When you are configuring Junos OS, your current hierarchy level is shown in the banner on the line preceding the **user@host#** prompt.

For a list of the complete configuration statement hierarchy, see the *Junos OS Configuration Statements and Commands*.

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


Copyright © 2012, Juniper Networks, Inc.

```
        clients {
          address <restrict>;
        }
      }
      routing-instance routing-instance-name {
        clients {
          address <restrict>;
        }
      }
      view view-name;
    }
    contact contact;
    description description;
    engine-id {
      (local engine-id | use-default-ip-address | use-mac-address);
    }
    filter-duplicates;
    interface [ interface-names ];
    location location;
    name name;
    nonvolatile {
      commit-delay seconds;
    }
    rmon {
      alarm index {
        description description;
        falling-event-index index;
        falling-threshold integer;
        falling-threshold-interval seconds;
        interval seconds;
        request-type (get-next-request | get-request | walk-request);
        rising-event-index index;
        rising-threshold integer;
        sample-type type;
        startup-alarm alarm;
        syslog-subtag syslog-subtag;
        variable oid-variable;
      }
      event index {
        community community-name;
        description description;
        type type;
      }
    }
    traceoptions {
      file filename <files number> <size size> <world-readable | no-world-readable> <match
        regular-expression>;
      flag flag;
    }
    trap-group group-name {
      categories {
        category;
      }
      destination-port port-number;
      routing-instance instance;
      logical-system logical-system-name;
```

```
    targets {
      address;
    }
    version (all | v1 | v2);
  }
  trap-options {
    agent-address outgoing-interface;
    source-address address;
    enterprise-oid;
    logical-system logical-system-name {
      routing-instance routing-instance-name {
        source-address address;
      }
    }
    routing-instance routing-instance-name {
      source-address address;
    }
  }
  v3 {
    notify name {
      tag tag-name;
      type (trap | inform);
    }
    notify-filter profile-name {
      oid oid (include | exclude);
    }
    snmp-community community-index {
      community-name community-name;
      security-name security-name;
      tag tag-name;
    }
    target-address target-address-name {
      address address;
      address-mask address-mask;
      logical-system logical-system;
      port port-number;
      retry-count number;
      routing-instance instance;
      tag-list tag-list;
      target-parameters target-parameters-name;
      timeout seconds;
    }
    target-parameters target-parameters-name {
      notify-filter profile-name;
      parameters {
        message-processing-model (v1 | v2c | v3);
        security-level (authentication | none | privacy);
        security-model (usm | v1 | v2c);
        security-name security-name;
      }
    }
    usm {
      local-engine {
        user username {
          authentication-md5 {
            authentication-password authentication-password;
```


Copyright © 2012, Juniper Networks, Inc.

```
                }
                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-prefiix){
                  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 13
- Configuring SNMP on a Device Running Junos OS on page 48

Copyright © 2012, Juniper Networks, Inc.

PART 2

Simple Network Management Protocol (SNMP)

- SNMP Overview on page 13
- Configuring SNMP on page 47
- SNMPv3 Overview on page 73
- Configuring SNMPv3 on page 75
- Standard SNMP Traps on page 113
- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Summary of SNMP Configuration Statements on page 145
- SNMP Remote Operations on page 169
- SNMP Support for Routing Instances on page 181

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 3

SNMP Overview

- Understanding the SNMP Implementation in Junos OS on page 13
- Standard SNMP MIBs Supported by Junos OS on page 17
- Juniper Networks Enterprise-Specific MIBs on page 31
- Loading MIB Files to a Network Management System on page 43

Understanding the SNMP Implementation in Junos OS

SNMP enables the monitoring of network devices from a central location. This topic provides an overview of SNMP and describes how SNMP is implemented in the Junos OS.

This topic includes the following sections:

- SNMP Architecture on page 13
- Junos OS SNMP Agent Features on page 16

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.

Copyright © 2012, Juniper Networks, Inc.

This topic contains the following sections:

- SNMP MIBs on page 14
- SNMP Traps and Informs on page 14

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

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

For a list of Juniper Networks enterprise-specific supported MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 31 in the *Junos OS SNMP MIBs and Traps Reference*.

SNMP Traps and Informs

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

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

For more information about standard traps supported by the Junos OS, see "Standard SNMP Traps Supported on Devices Running Junos OS" on page 113 in the *Junos OS SNMP MIBs and Traps Reference*.

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

 Copyright © 2012, Juniper Networks, Inc.

For more information about enterprise-specific traps supported by the Junos OS, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 125 in the *Junos OS SNMP MIBs and Traps Reference*. For information about system logging severity levels for SNMP traps, see "System Logging Severity Levels for SNMP Traps" on page 15.

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 101.

SNMP Trap Queuing

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

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

The Junos OS also has a throttle mechanism to control the number of traps (**throttle threshold**; default value of 500 traps) sent during a particular time period (**throttle interval**; default of 5 seconds) and to ensure consistency in trap traffic, especially when a large number of traps are generated because of interface status changes. The throttle interval period begins when the first trap arrives at the throttle. All traps within the trap threshold are processed, and the traps beyond the threshold limit are queued. The maximum size of trap queues (that is, the throttle queue and the destination queue combined) is 40,000 traps. However, on EX Series switches, the maximum size of the trap queue is 1000 traps. The maximum size of any one queue is 20,000 traps for devices other than EX Series switches. On EX Series switches, the maximum size of one queue is 500 traps. If a trap is sent from a destination queue when the throttle queue has exceeded the maximum size, the trap is added back to the top of the destination queue, and all subsequent attempts from the destination queue are stopped for a 30-second period, after which the destination queue restarts sending the traps.

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

System Logging Severity Levels for SNMP Traps

For some traps, when a trap condition occurs, regardless of whether the SNMP agent sends a trap to an NMS, the trap is logged if the system logging is configured to log an

event with that system logging severity level. For more information about system logging severity levels, see the *Junos OS System Basics Configuration Guide*.

For more information about system logging severity levels for standard traps, see "Standard SNMP Version 1 Traps" on page 113 and "Standard SNMP Version 2 Traps" on page 117 in the *Junos OS SNMP MIBs and Traps Reference*. For more information about system logging severity levels for enterprise-specific traps, see "Juniper Networks Enterprise-Specific SNMP Version 1 Traps" on page 126 and "Juniper Networks Enterprise-Specific SNMP Version 2 Traps" on page 133 in the *Junos OS SNMP MIBs and Traps Reference*.

Junos OS SNMP Agent Features

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

The Junos OS supports the following versions of SNMP:

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

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

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

Related Documentation

- SNMPv3 Overview on page 73
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

 Copyright © 2012, Juniper Networks, Inc.

Standard SNMP MIBs Supported by Junos OS

Table 4 on page 17 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 (M, T, J, MX, EX, 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							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
IEEE 802.1ab section 12.1, *Link Layer Discovery Protocol (LLDP) MIB* Supported tables and objects: • **lldpRemManAddrOID**, **lldpLocManAddrOID**, **lldpReinitDelay**, **lldpNotificationInterval**, **lldpStatsRxPortFramesDiscardedTotal**, **lldpStatsRxPortFramesError**, **lldpStatsRxPortTLVsDiscardedTotal**, **lldpStatsRxPortTLVsUnrecognizedTotal**, **lldpStatsRxPortAgeoutsTotal**	0	0	0	0	1	0		

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
IEEE, 802.3ad, *Aggregation of Multiple Link Segments* Supported tables and objects: • **dot3adAggPortTable**, **dot3adAggPortListTable**, **dot3adAggTable**, and **dot3adAggPortStatsTable** NOTE: EX Series switches do not support the **dot3adAggPortTable** and **dot3adAggPortStatsTable**. • **dot3adAggPortDebugTable** (only **dot3adAggPortDebugRxState**, **dot3adAggPortDebugMuxState**, **dot3adAggPortDebugActorSyncTransitionCount**, **dot3adAggPortDebugPartnerSyncTransitionCount**, **dot3adAggPortDebugActorChangeCount**, and **dot3adAggPortDebugPartnerChangeCount**) NOTE: EX Series switches do not support the **dot3adAggPortDebugTable**. • **dot3adTablesLastChanged** NOTE: Gigabit Ethernet interfaces on J Series Services Routers do not support the 802.3ad MIB.	1	1	1	1	1	1	1	1
RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*	1	1	1	1	1	1	1	1
RFC 1157, *A Simple Network Management Protocol (SNMP)*	1	1	1	1	1	1	1	1
RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments* (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
RFC 1212, *Concise MIB Definitions*	1	1	1	1	1	0	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End

RFC 1213, *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II*. Junos OS supports the following areas:
• MIB II and its SNMP version 2 derivatives, including:
  • Statistics counters
  • IP, except for **ipRouteTable**, which has been replaced by **ipCidrRouteTable** (RFC 2096, *IP Forwarding Table MIB*)
  • SNMP management
  • Interface management
• SNMPv1 **Get**, **GetNext** requests, and version 2 **GetBulk** request
• Junos OS-specific secured access list
• Master configuration keywords
• Reconfigurations upon SIGHUP
	1	1	1	1	1	0	0	1

RFC 1215, *A Convention for Defining Traps for use with the SNMP* (only MIB II SNMP version 1 traps and version 2 notifications)	1	1	1	1	1	0	0	1
RFC 1406, *Definitions of Managed Objects for the DS1 and E1 Interface Types* (T1 MIB is supported)	1	1	1	0	0	1	0	0
RFC 1407, *Definitions of Managed Objects for the DS3/E3 Interface Type* (T3 MIB is supported)	1	1	1	0	0	0	0	0
RFC 1471, *Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol* (only pppLink group is supported. The pppLink group consists of the **pppLcp 1** object and the tables **pppLinkStatustable** and **pppLinkConfigTable**).	1	0	0	1	0	0	0	0
RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*	1	1	1	1	1	0	0	0
RFC 1695, *Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2*	1	1	1	0	0	0	0	0

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 1850, *OSPF Version 2 Management Information Base* (except for the **ospfOriginateNewLsas** and **ospfRxNewLsas** objects, the Host Table, and the traps **ospfOriginateLSA**, **ospfLsdbOverflow**, and **ospfLsdbApproachingOverflow**)	1	1	1	1	1	1	0	0
RFC 1901, *Introduction to Community-based SNMPv2*	1	1	1	1	1	1	1	1
RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2*	1	1	1	1	1	0	0	0
RFC 2012, *SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2*	1	1	1	1	1	1	0	1
RFC 2013, *SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2*	1	1	1	1	1	1	0	1
RFC 2024, *Definitions of Managed Objects for Data Link Switching Using SMIv2* (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)	1	1	1	1	0	0	0	0
RFC 2096, *IP Forwarding Table MIB* (The **ipCidrRouteTable** has been extended to include the tunnel name when the next hop is through an RSVP-signaled LSP.)	1	1	1	1	1	0	0	1
RFC 2115, *Management Information Base for Frame Relay DTEs Using SMIv2* (**frDlcmiTable** only; **frCircuitTable** and **frErrTable** are not supported)	1	1	1	1	0	1	0	0
RFC 2233, *The Interfaces Group MIB Using SMIv2* NOTE: RFC 2233 has been replaced by RFC 2863, IF MIB. However, Junos OS supports both RFC 2233 and RFC 2863.	1	1	1	1	1	1	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 2287, *Definitions of System-Level Managed Objects for Applications* (only the objects **sysApplInstallPkgTable**, **sysApplInstallElmtTable**, **sysApplElmtRunTable**, and **sysApplMapTable**)	1	1	1	1	1	1	0	1
RFC 2465, *Management Information Base for IP Version 6: Textual Conventions and General Group* (except for IPv6 interface statistics)	1	1	1	1	0	1	0	0
RFC 2495, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types* (except for **dsx1FarEndConfigTable**, **dsx1FarEndCurrentTable**, **dsx1FarEndIntervalTable**, **dsx1FarEndTotalTable**, and **dsx1FracTable**)	1	1	1	0	0	1	0	0
RFC 2515, *Definitions of Managed Objects for ATM Management* (except **atmVpCrossConnectTable**, **atmVcCrossConnectTable**, and **aal5VccTable**)	1	1	1	0	0	0	0	0
RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework*	1	1	1	1	1	0	0	1
RFC 2571, *An Architecture for Describing SNMP Management Frameworks* (read-only access) NOTE: RFC 2571 has been replaced by RFC 3411. However, Junos OS supports both RFC 2571 and RFC 3411.	1	1	1	1	1	1	0	1
RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* (read-only access) NOTE: RFC 2572 has been replaced by RFC 3412. However, Junos OS supports both RFC 2572 and RFC 3412.	1	1	1	1	1	1	0	1
RFC 2576, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework* NOTE: RFC 2576 has been replaced by RFC 3584. However, Junos OS supports both RFC 2576 and RFC 3584.	1	1	1	1	1	1	0	1
RFC 2578, *Structure of Management Information Version 2 (SMIv2)*	1	1	1	1	1	0	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 2579, *Textual Conventions for SMIv2*	1	1	1	1	1	0	0	1
RFC 2580, *Conformance Statements for SMIv2*	1	1	1	1	1	0	0	1
RFC 2662, *Definitions of Managed Objects for ADSL Lines* (J Series Services Routers. All MIB tables, objects, and traps are applicable for the ADSL ATU-R agent.)	1	1	1	1	0	1	0	0
RFC 2665, *Definitions of Managed Objects for the Ethernet-like Interface Types*	1	1	1	1	1	1	0	1
RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol* (except row creation, the **Set** operation, and the object **vrrpStatsPacketLengthErrors**)	1	1	1	1	1	1	0	1
RFC 2790, *Host Resources MIB* • 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.	1	1	1	1	1	1	0	1
RFC 2819, *Remote Network Monitoring Management Information Base* • **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.	1	1	1	1	1	1	0	1
RFC 2863, *The Interfaces Group MIB* NOTE: RFC 2863 replaces RFC 2233. However, Junos OS supports both RFC 2233 and RFC 2863.	1	1	1	1	1	0	0	1
RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*	1	1	1	1	0	0	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 2922, *The Physical Topology (PTOPO) MIB* Supported objects: **ptopoConnDiscAlgorithm, ptopoConnAgentNetAddrType, ptopoConnAgentNetAddr, ptopoConnMultiMacSASeen, ptopoConnMultiNetSASeen, ptopoConnIsStatic, ptopoConnLastVerifyTime, ptopoConnRowStatus**	0	0	0	0	1	1	0	1
RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations* (only the objects **pingCtlTable**, **pingResultsTable**, **pingProbeHistoryTable**, **pingMaxConcurrentRequests**, **traceRouteCtlTable**, **traceRouteResultsTable**, **traceRouteProbeHistoryTable**, and **traceRouteHopsTable**)	1	1	1	1	1	1	0	1
RFC 2932, *IPv4 Multicast Routing MIB*	1	1	1	1	1	1	0	1
RFC 2933, *Internet Group Management Protocol (IGMP) MIB*	1	1	1	1	1	1	0	0
RFC 2934, *Protocol Independent Multicast MIB for IPv4* NOTE: In Junos OS, RFC 2934 is implemented based on a draft version, *pimmib.mib*, of the now standard RFC. Support for the **pimNeighborLoss** trap was added in Release 11.4.	1	1	1	1	1	1	0	0
RFC 2981, *Event MIB*	1	1	1	1	0	0	0	0
RFC 3014, *Notification Log MIB*	1	1	1	1	0	0	0	0
RFC 3019, *IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol*	1	1	1	1	0	0	0	1
RFC 3410 *Introduction and Applicability Statements for Internet-Standard Management Framework*	1	1	1	1	1	0	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks* NOTE: RFC 3411 replaces RFC 2571. However, Junos OS supports both RFC 3411 and RFC 2571.	1	1	1	1	1	0	0	1
RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* NOTE: RFC 3412 replaces RFC 2572. However, Junos OS supports both RFC 3412 and RFC 2572.	1	1	1	1	1	0	0	1
RFC 3413, *Simple Network Management Protocol (SNMP) Applications* (except for the Proxy MIB)	1	1	1	1	1	1	0	1
RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*	1	1	1	1	1	0	0	1
RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*	1	1	1	1	1	0	0	1
RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)* NOTE: RFC 3416 replaces RFC 1905, which was supported in earlier versions of Junos OS.	1	1	1	1	1	0	0	1
RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)*	1	1	1	1	1	1	0	1
RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)* NOTE: RFC 3418 replaces RFC 1907, which was supported in earlier versions of Junos OS.	1	1	1	1	1	0	0	1
RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures* (implemented under the Juniper Networks enterprise branch [**jnxExperiment**])	1	1	1	0	0	0	0	0

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 3584 *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*	1	1	1	1	1	0	0	1
RFC 3591 *Managed Objects for the Optical Interface Type* **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**)	1	1	1	0	0	0	0	0
RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*	1	1	1	1	0	0	0	0
RFC 3621, *Power Ethernet MIB*	0	0	0	0	1	0	0	0
RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer* (except **etherWisDeviceTable**, **etherWisSectionCurrentTable**, and **etherWisFarEndPathCurrentTable**)	1	1	1	1	0	0	0	0
RFC 3811, *Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management*	1	1	1	1	0	1	0	0

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 3812, *Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)* (read only access) • 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. 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**.	1	1	1	1	0	0	0	0
RFC 3813, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)* (read-only access). **mplsInterfacePerfTable**, **mplsInSegmentPerfTable**, **mplsOutSegmentPerfTable**, **mplsInSegmentMapTable**, **mplsXCUp**, and **mplsXCDown** are not supported.	1	1	1	1	0	1	0	0
RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*	1	1	1	1	1	0	0	1
RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type* (except **dsx3FarEndConfigTable**, **dsx3FarEndCurrentTable**, **dsx3FarEndIntervalTable**, **dsx3FarEndTotalTable**, and **dsx3FracTable**)	1	1	1	0	0	0	0	0

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 4188, *Definitions of Managed Objects for Bridges*—Supports 802.1D STP(1998). Supports only the following subtrees and objects: • **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. NOTE: **dot1dTpLearnedEntryDiscards** and **dot1dTpAgingTime** objects are supported on M and T Series routers.	0	0	0	1	1	0	0	0
RFC 4318, *Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol*—Supports 802.1w and 802.1t extensions for RSTP.	1	1	1	1	1	0	0	0
RFC 4363b, *Q-Bridge VLAN MIB*	0	0	0	1	1	0	0	0
RFC 4382 *MPLS/BGP Layer 3 Virtual Private Network (VPN) MIB* The Junos OS support for RFC 4382 includes the following scalar objects and tables: • **mplsL3VpnConfiguredVrfs** • **mplsL3VpnActiveVrfs** • **mplsL3VpnConnectedInterfaces** • **mplsL3VpnNotificationEnable** • **mplsL3VpnVrfConfMaxPossRts** • **mplsL3VpnVrfConfRteMxThrshTime** • **mplsL3VpnIllLblRcvThrsh** • **mplsL3VpnVrfTable** • **mplsL3VpnIfConfTable** • **mplsL3VpnVrfPerfTable** • **mplsL3VpnVrfRteTable** • **mplsVpnVrfRTTable**	1	1	1	1	1	0	0	0
RFC 4444, *IS-IS MIB*	1	1	1	1	1	1	0	0
RFC 4668, *RADIUS Accounting Client Management Information Base (MIB) for IPv6* (read-only access)	0	0	0	1	0	0	0	0

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 4670, *RADIUS Accounting Client Management Information Base (MIB)* (read-only access)	0	0	0	1	0	0	0	0
RFC 4801, *Definitions of Textual Conventions for Generalized Multiprotocol Label Switching (GMPLS) Management Information Base (MIB)* (read-only access)	1	1	1	1	0	0	0	0
RFC 4802, *Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering (TE) Management Information Base (MIB)* (read-only access). **gmplsTunnelReversePerfTable**, **gmplsTeScalars**, **gmplsTunnelTable**, **gmplsTunnelARHopTable**, **gmplsTunnelCHopTable**, and **gmplsTunnelErrorTable** are not supported.)	1	1	1	1	0	0	0	0
RFC 4803, *Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) Management Information Base (MIB)* (read-only access). **gmplsLabelTable** and **gmplsOutsegmentTable** are not supported. 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.	1	1	1	1	0	0	0	0

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
RFC 5643, *Management Information Base for OSPFv3* NOTE: Junos OS support for this MIB is read-only. Junos OS does not support the following tables and objects defined in this MIB. • **ospfv3HostTable** • **ospfv3CfgNbrTable** • **ospfv3ExitOverflowInterval** • **ospfv3ReferenceBandwidth** • **ospfv3RestartSupport** • **ospfv3RestartInterval** • **ospfv3RestartStrictLsaChecking** • **ospfv3RestartStatus** • **ospfv3RestartAge** • **ospfv3RestartExitReason** • **ospfv3NotificationEnable** • **ospfv3StubRouterSupport** • **ospfv3StubRouterAdvertisement** • **ospfv3DiscontinuityTime** • **ospfv3RestartTime** • **ospfv3AreaNssaTranslatorRole** • **ospfv3AreaNssaTranslatorState** • **ospfv3AreaNssaTranslatorStabInterval** • **ospfv3AreaNssaTranslatorEvents** • **ospfv3AreaTEEnabled** • **ospfv3IfMetricValue** • **ospfv3IfDemandNbrProbe**	1	1	1	1	0	0	0	1
Internet Assigned Numbers Authority, *IANAiftype Textual Convention MIB* (referenced by RFC 2233, available at http://www.iana.org/assignments/ianaiftype-mib)	1	1	1	1	1	1	0	0
Internet draft draft-ietf-atommib-sonetaps-mib-10.txt, *Definitions of Managed Objects for SONET Linear APS Architectures* (as defined under the Juniper Networks enterprise branch [**jnxExperiment**] only)	1	1	1	1	0	0	0	0

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
Internet draft draft-ieft-bfd-mib-02.txt, *Bidirectional Forwarding Detection Management Information Base* (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	0	0	1
Internet draft draft-ietf-idmr-igmp-mib-13.txt, *Internet Group Management Protocol (IGMP) MIB*	1	1	1	1	1	0	0	1
Internet draft draft-ietf-idr-bgp4-mibv2-04.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version* (only **jnxBgpM2PrefixInPrefixes**, **jnxBgpM2PrefixInPrefixesAccepted**, and **jnxBgpM2PrefixInPrefixesRejected** objects)	1	1	1	1	1	0	0	1
Internet draft draft-reeder-snmpv3-usm-3desede-00.txt, *Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode*	1	1	1	1	1	0	0	1
Internet draft draft-ietf-isis-wg-mib-07.txt, *Management Information Base for IS-IS* (only **isisISAdjTable**, **isisISAdjAreaAddrTable**, **isisISAdjIPAddrTable**, and **isisISAdjProtSuppTable)** NOTE: Replaced with RFC 4444, *IS-IS MIB* in Junos OS Release 11.3 and later.	1	1	1	1	1	1	0	0
Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt, *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (only **mplsVpnScalars**, **mplsVpnVrfTable,** **mplsVpnPerTable**, and **mplsVpnVrfRouteTargetTable)**	1	1	1	1	0	0	0	0
Internet draft draft-ietf-ospf-ospfv3-mib-11.txt, *Management Information Base for OSPFv3* (Represented by **mib-jnx-ospfv3mib.txt** and implemented under the Juniper Networks enterprise branch **{jnxExperiment}**. Support for **ospfv3NbrTable** only. Read only. Object names are prefixed by **jnx**. For example, **jnxOspfv3NbrTable**, **jnxOspfv3NbrAddressType**, and **jnxOspfv3NbrPriority**.)	1	1	1	1	0	0	0	1

Copyright © 2012, Juniper Networks, Inc.

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

MIB/RFC	Platforms							
						SRX		
	M	T	J	MX	EX	Low-End	Mid-Range	High-End
Internet draft draft-ietf-idmr-pim-mib-09.txt, *Protocol Independent Multicast (PIM) MIB*	1	1	1	1	1	0	0	1
ESO Consortium MIB, which can be found at http://www.snmp.com/eso/ NOTE: The ESO Consortium MIB has been replaced by RFC 3826.	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 **mplsTeP2mpTunnelBranchPerfTable**).	1	1	1	1	0	0	0	0

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Loading MIB Files to a Network Management System on page 43

Juniper Networks Enterprise-Specific MIBs

The 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/junos12.1/topics/reference/mibs/mib-jnx-user-aaa.txt

  For more information, see "AAA Objects MIB" on page 213.

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-auth.txt.

  For more information, see "Access Authentication Objects MIB" on page 219.

- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-chassis-alarm.txt .

  For more information, see "Alarm MIB" on page 223.

Copyright © 2012, Juniper Networks, Inc.

- 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/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-atm-cos.txt

  For more information, see "ATM Class-of-Service MIB" on page 227.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-bgpmib2.txt .

  For more information, see "BGP4 V2 MIB" on page 235.

- 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/junos12.1/topics/ reference/mibs/mib-jnx-bfd.txt .

  For more information, see "Bidirectional Forwarding Detection MIB" on page 237.

- 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/junos12.1/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

 Copyright © 2012, Juniper Networks, Inc.

forwarding process (**fwdd**) components. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-jsrpd.txt .

  For more information, see "Chassis Cluster MIB" on page 365.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-cfgmgmt.txt .

  For more information, see "Configuration Management MIB" on page 369.

- 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/junos12.1/topics/reference/mibs/mib-jnx-dcu.txt .

  For more information, see "Destination Class Usage MIB" on page 375.

- 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/junos12.1/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/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-js-dns.txt .

  For more information, see "DNS Objects MIB" on page 379.

- 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/junos12.1/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/junos12.1/topics/ reference/mibs/jnx-mac.txt .

  For more information, see "Ethernet MAC MIB" on page 381.

- 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/junos12.1/topics/reference/mibs/mib-jnx-event.txt .

  For more information, see "Event MIB" on page 383.

- Experimental MIB—Contains object identifiers for experimental MIBs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-exp.txt .

  For more information, see "jnxExperiment" on page 212.

Copyright © 2012, Juniper Networks, Inc.

- 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/junos12.1/topics/reference/mibs/mib-jnx-firewall.txt .

  For more information, see "Firewall MIB" on page 385.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-hostresources.txt .

  For more information, see "Host Resources MIB" on page 389.

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-idp.txt .

  For more information, see "IDP MIB" on page 537.

- 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/junos12.1/topics/reference/mibs/mib-jnx-if-extensions.txt .

  For more information, see "Interface MIB" on page 391.

- IP Forward MIB—Extends the standard IP Forwarding Table MIB (RFC 2096) to include CIDR forwarding information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-ipforward.txt .

  For more information, see "IP Forward MIB" on page 397.

- 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/junos12.1/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt .

For more information, see "IPsec Generic Flow Monitoring Object MIB" on page 399.

- 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/junos12.1/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt .

  For more information, see "IPSec Monitoring MIB" on page 415.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-ipv4.txt .

  For more information, see "IPv4 MIB" on page 423.

- 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/junos12.1/topics/reference/mibs/mib-jnx-ipv6.txt .

  For more information, see "IPv6 MIB" on page 425.

- 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/junos12.1/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/junos12.1/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

 Copyright © 2012, Juniper Networks, Inc.

http://www.juniper.net/techpubs/en_US/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-license.txt .

  For more information, see "License MIB" on page 433.

- Logical Systems MIBs—Extend SNMP support to logical systems security profile through various MIBs defined under **jnxLsysSecurityProfile**.

  For more information about logical systems MIBs and downloadable versions of the MIBs, see "Logical Systems MIB" on page 437.

- 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/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-mpls-ldp.txt .

NOTE: Objects in the MPLS LDP MIB were supported in earlier releases of the 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/junos12.1/topics/reference/mibs/mib-jnx-js-nat.txt .

  For more information, see "NAT Objects MIB" on page 479.

- 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/junos12.1/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 the Junos OS. For a downloadable version of the MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/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/junos12.1/topics/ reference/mibs/mib-jnx-pfe.txt .

  For more information, see "Packet Forwarding Engine MIB" on page 487.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-pae-extension.txt .

Copyright © 2012, Juniper Networks, Inc.

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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-ping.txt .

  For more information, see "PING MIB" on page 491.

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-policy.txt .

  For more information, see "Policy Objects MIB" on page 507.

- Power Supply Unit MIB—Enables monitoring and managing of the power supply on a device running the 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/junos12.1/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 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/junos12.1/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 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-pwtdm.txt .

- 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/junos12.1/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/junos12.1/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" on page 513.

- 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/junos12.1/topics/reference/mibs/mib-jnx-rmon.txt .

  For more information, see "RMON Events and Alarms MIB" on page 515.

- 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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-js-if-ext.txt .

  For more information, see "Security Interface Extension Objects MIB" on page 519.

Copyright © 2012, Juniper Networks, Inc.

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-screening.txt .

  For more information, see "Security Screening Objects MIB" on page 523.

- 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/junos12.1/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/junos12.1/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/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-scu.txt .

  For more information, see "Source Class Usage MIB" on page 539.

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt .

  For more information, see "SPU Monitoring Objects MIB" on page 541.

- 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/junos12.1/topics/reference/mibs/mib-jnx-smi.txt .

  For more information, see "Structure of Management Information MIB" on page 207.

- 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/junos12.1/topics/reference/mibs/mib-jnx-ex-smi.txt .

Copyright © 2012, Juniper Networks, Inc.

For more information, see EX Series SMI MIB.

- Structure of Management Information MIB—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/junos12.1/topics/reference/mibs/mib-jnx-js-smi.txt .

- Subscriber MIB—Provides SNMP support for subscriber-related information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/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/junos12.1/topics/reference/mibs/mib-jnx-syslog.txt .

  For more information, see "System Log MIB" on page 545.

- 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/junos12.1/topics/reference/mibs/mib-jnx-traceroute.txt .

  For more information, see "Traceroute MIB" on page 549.

- Utility MIB—Provides SNMP support for exposing 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/junos12.1/topics/ reference/mibs/mib-jnx-util.txt .

  For more information, see "Utility MIB" on page 551.

- Virtual Chassis MIB—Contains information about the virtual chassis on the EX Series Ethernet Switches. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-virtualchassis.txt .

  For more information, see Virtual Chassis MIB.

- 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/junos12.1/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

 Copyright © 2012, Juniper Networks, Inc.

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/junos12.1/topics/reference/mibs/mib-jnx-vpls-generic.txt
- http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-vpls-bgp.txt
- http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-vpls-ldp.txt

- 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/junos12.1/topics/reference/mibs/mib-jnx-js-cert.txt .

  For more information, see "VPN Certificate Objects MIB" on page 555.

- 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/junos12.1/topics/reference/mibs/mib-jnx-vpn.txt .

  For more information, see VPN MIB.

Related Documentation

- Standard SNMP MIBs Supported by Junos OS on page 17
- Enterprise-Specific MIBs and Supported Devices
- Loading MIB Files to a Network Management System on page 43

Loading MIB Files to a Network Management System

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

You can download the Junos MIB package from the **Enterprise-Specific MIBs and Traps** section of the Junos OS Technical Publications index page at http://www.juniper.net/techpubs/software/junos/index.html . The Junos MIB package is

available in **.zip** and **.tar** packages. You can download the appropriate format based on your requirements.

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

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

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

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

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

Copyright © 2012, Juniper Networks, Inc.

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

8. Load the Juniper Networks enterprise-specific SMI MIB, **mib-jnx-smi.txt**, and the following optional SMI MIBs based on your requirements:
   - **mib-jnx-js-smi.txt**—(Optional) For Juniper Security MIB tree objects
   - **mib-jnx-ex-smi.txt**—(Optional) For EX Series Ethernet Switches
   - **mib-jnx-exp.txt**—(Recommended) For Juniper Networks experimental MIB objects
9. Load the remaining enterprise-specific MIBs from the **JuniperMibs** folder.

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

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

Related Documentation

- Standard SNMP MIBs Supported by Junos OS on page 17
- Juniper Networks Enterprise-Specific MIBs on page 31

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 4

Configuring SNMP

- Configuring SNMP on a Device Running Junos OS on page 48
- Configuring the System Contact on a Device Running Junos OS on page 50
- Configuring the System Location for a Device Running Junos OS on page 50
- Configuring the System Description on a Device Running Junos OS on page 51
- Filtering Duplicate SNMP Requests on page 51
- Configuring the Commit Delay Timer on page 52
- Configuring the System Name on page 52
- Examples: Configuring the SNMP Community String on page 53
- Example: Configuring SNMP Trap Groups on page 54
- Configuring the SNMP Community String on page 54
- Adding a Group of Clients to an SNMP Community on page 55
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Configuring SNMP Trap Options on page 57
- Configuring SNMP Trap Groups on page 60
- Spoofing Enterprise-Specific SNMP Traps on page 63
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 63
- Example: Configuring Secured Access List Checking on page 64
- Example: Ping Proxy MIB on page 64
- Filtering Interface Information Out of SNMP Get and GetNext Output on page 65
- Example: Tracing SNMP Activity on page 65
- Configuring MIB Views on page 66
- Tracing SNMP Activity on a Device Running Junos OS on page 67
- Configuring the Local Engine ID on page 70

Copyright © 2012, Juniper Networks, Inc.

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 ];
```


Copyright © 2012, Juniper Networks, Inc.

```
    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 13
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Complete SNMPv3 Configuration Statements on page 76

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 48
- Configuring the System Location for a Device Running Junos OS on page 50
- Configuring the System Description on a Device Running Junos OS on page 51
- Configuring the System Name on page 52
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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 48
- Configuring the System Contact on a Device Running Junos OS on page 50
- Configuring the System Description on a Device Running Junos OS on page 51
- Configuring the System Name on page 52
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

Copyright © 2012, Juniper Networks, Inc.

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 48
- Configuring the System Contact on a Device Running Junos OS on page 50
- Configuring the System Location for a Device Running Junos OS on page 50
- Configuring the System Name on page 52
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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 48
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 63
- Filtering Interface Information Out of SNMP Get and GetNext Output on page 65
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

Configuring the Commit Delay Timer

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

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

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

seconds is the length of the time between when the SNMP request is received and the commit is requested for the candidate configuration. For more information about the **configure exclusive** command and locking the configuration, see the *Junos OS CLI User Guide*.

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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 "snmp 1";
}
```

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 48
- Configuring the System Contact on a Device Running Junos OS on page 50
- Configuring the System Location for a Device Running Junos OS on page 50
- Configuring the System Description on a Device Running Junos OS on page 51

Copyright © 2012, Juniper Networks, Inc.

- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

Examples: Configuring the SNMP Community String

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

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

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

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

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

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

Related Documentation

- Configuring the SNMP Community String on page 54

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 60
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Configuring SNMP Trap Options on page 57

Configuring the SNMP Community String

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

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

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

The default authorization level for a community is **read-only**. To allow **Set** requests within a community, you need to define that community as **authorization read-write**. For **Set** requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the **view** statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information about the **view** statement, see "Configuring MIB Views" on page 66.

Copyright © 2012, Juniper Networks, Inc.

The **clients** statement lists the IP addresses of the clients (community members) that are allowed to use this community. If no **clients** statement is present, all clients are allowed. For ***address***, you must specify an IPv4 or IPv6 address, not a hostname. Include the **default restrict** option to deny access to all SNMP clients for which access is not explicitly granted. We recommend that you always include the **default restrict** option to limit SNMP client access to the local router.

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

Related Documentation

- Adding a Group of Clients to an SNMP Community on page 55
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Examples: Configuring the SNMP Community String on page 53

Adding a Group of Clients to an SNMP Community

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

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

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

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

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

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

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

The following example shows how to define a client list:

```
[edit]
```

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

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

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

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

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

Related Documentation

- client-list
- client-list-name

Configuring 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.

 Copyright © 2012, Juniper Networks, Inc.

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 57
- Configuring SNMP Trap Groups on page 60
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

Configuring SNMP Trap Options

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

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

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

```
[edit snmp]
trap-options {
   agent-address outgoing-interface;
   enterprise-oid
   logical-system
   routing-instance
   source-address address;
}
```

You must also configure a trap group for the trap options to take effect. For information about trap groups, see "Configuring SNMP Trap Groups" on page 60.

This topic contains the following sections:

- Configuring the Source Address for SNMP Traps on page 58
- Configuring the Agent Address for SNMP Traps on page 59
- Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps on page 60

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 {
```

 Copyright © 2012, Juniper Networks, Inc.

```
    source-address lo0;
  }
  trap-group "urgent-dispatcher" {
    version v2;
    categories link startup;
    targets {
      192.168.10.22;
      172.17.1.2;
    }
  }
  [edit interfaces]
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.1/32;
        address 127.0.0.1/32;
      }
    }
  }
```

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

Logical System Name as the Source Address

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

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

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

Routing Instance Name as the Source Address

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

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

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

Configuring the Agent Address for SNMP Traps

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

```
[edit snmp]
trap-options {
```

```
    agent-address outgoing-interface;
  }
```

To configure the outgoing interface as the agent address:

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

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

Adding snmpTrapEnterprise Object Identifier to Standard SNMP Traps

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

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

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

Related Documentation

- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Configuring SNMP Trap Groups on page 60
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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 {
```

 Copyright © 2012, Juniper Networks, Inc.

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

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

For each trap group that you define, you must include the **target** statement to define at least one system as the recipient of the SNMP traps in the trap group. Specify the IPv4 or IPv6 address of each recipient, not its hostname.

Specify the types of traps the trap group can receive in the **categories** statement. For information about the category to which the traps belong, see the "Standard SNMP Traps Supported on Devices Running Junos OS" on page 113 and "Juniper Networks Enterprise-Specific SNMP Traps" on page 125 topics in the *Junos OS SNMP MIBs and Traps Reference*.

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

A trap group can receive the following categories:

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

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

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

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

  - **loss-of-light**—Loss of light alarm notification
  - **pll-lock**—PLL lock alarm notification
  - **loss-of-frame**—Loss of frame alarm notification
  - **loss-of-signal**—Loss of signal alarm notification
  - **severely-errored-frame**—Severely errored frame alarm notification
  - **line-ais**—Line alarm indication signal (AIS) alarm notification
  - **path-ais**—Path AIS alarm notification
  - **loss-of-pointer**—Loss of pointer alarm notification
  - **ber-defect**—SONET/SDH bit error rate alarm defect notification
  - **ber-fault**—SONET/SDH error rate alarm fault notification
  - **line-remote-defect-indication**—Line remote defect indication alarm notification
  - **path-remote-defect-indication**—Path remote defect indication alarm notification
  - **remote-error-indication**—Remote error indication alarm notification
  - **unequipped**—Unequipped alarm notification
  - **path-mismatch**—Path mismatch alarm notification
  - **loss-of-cell**—Loss of cell delineation alarm notification
  - **vt-ais**—Virtual tributary (VT) AIS alarm notification
  - **vt-loss-of-pointer**—VT loss of pointer alarm notification
  - **vt-remote-defect-indication**—VT remote defect indication alarm notification
  - **vt-unequipped**—VT unequipped alarm notification
  - **vt-label-mismatch**—VT label mismatch error notification
  - **vt-loss-of-cell**—VT loss of cell delineation notification
- **startup**—System warm and cold starts
- **vrrp-events**—Virtual Router Redundancy Protocol (VRRP) events such as new-master or authentication failures

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

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

 Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Configuring SNMP Trap Options on page 57
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Example: Configuring SNMP Trap Groups on page 54

Spoofing Enterprise-Specific SNMP Traps

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

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

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 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Example: Configuring Secured Access List Checking on page 64
- 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 63
- Filtering Interface Information Out of SNMP Get and GetNext Output on page 65
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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;
}
```

 Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Configuring MIB Views on page 66
- **view (Configuring a MIB View) on page 168**
- **oid on page 157**

Filtering Interface Information Out of SNMP Get and GetNext Output

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

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

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

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

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

Related Documentation

- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 63
- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

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 48
- Tracing SNMP Activity on a Device Running Junos OS on page 67
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6

Configuring MIB Views

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

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

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

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

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

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

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

For more information about the Ping MIB, see RFC 2925 and the "PING MIB" on page 491 topic in the *Junos OS SNMP MIBs and Traps Reference*.

Related Documentation

- Configuring SNMP on a Device Running Junos OS on page 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Example: Ping Proxy MIB on page 64

Copyright © 2012, Juniper Networks, Inc.

-
- view (Configuring a MIB View) on page 168
- view (Associating MIB View with a Community)
- oid on page 157

Tracing SNMP Activity on a Device Running Junos OS

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

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

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

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

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

These statements are described in the following sections:

- Configuring the Number and Size of SNMP Log Files on page 68
- Configuring Access to the Log File on page 68
- Configuring a Regular Expression for Lines to Be Logged on page 68
- Configuring the Trace Operations on page 69

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;
```

 Copyright © 2012, Juniper Networks, Inc.

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 5 on page 69 describes the meaning of the SNMP tracing flags.

Table 5: 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
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

Copyright © 2012, Juniper Networks, Inc.

Table 5: SNMP Tracing Flags *(continued)*

Flag	Description	Default Setting
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 *agent*d | 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 48
- Configuration Statements at the [edit snmp] Hierarchy Level on page 6
- Example: Tracing SNMP Activity on page 65
- Configuring SNMP

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.

Copyright © 2012, Juniper Networks, Inc.

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 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: SNMPv3 Configuration on page 108

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 5

SNMPv3 Overview

- SNMPv3 Overview on page 73

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 80
- Configuring MIB Views on page 66

- Defining Access Privileges for an SNMP Group on page 84
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring SNMP Informs on page 101

Related Documentation

- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 6

Configuring SNMPv3

- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Configuring the Local Engine ID on page 79
- Creating SNMPv3 Users on page 80
- Configuring the SNMPv3 Authentication Type on page 81
- Configuring the Encryption Type on page 82
- Defining Access Privileges for an SNMP Group on page 84
- Configuring the Access Privileges Granted to a Group on page 85
- Example: Access Privilege Configuration on page 88
- Assigning Security Model and Security Name to a Group on page 89
- Example: Security Group Configuration on page 91
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring the SNMPv3 Trap Notification on page 93
- Example: Configuring SNMPv3 Trap Notification on page 93
- Configuring the Trap Notification Filter on page 94
- Configuring the Trap Target Address on page 95
- Example: Configuring the Tag List on page 98
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Configuring the Remote Engine and Remote User on page 102
- Example: Configuring the Remote Engine ID and Remote Users on page 103
- Configuring the Inform Notification Type and Target Address on page 104
- Example: Configuring the Inform Notification Type and Target Address on page 105
- Configuring the SNMPv3 Community on page 106
- Example: SNMPv3 Community Configuration on page 108
- Example: SNMPv3 Configuration on page 108

Copyright © 2012, Juniper Networks, Inc.

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 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;
```


Copyright © 2012, Juniper Networks, Inc.

```
            }
            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 80
- Configuring MIB Views on page 66
- Defining Access Privileges for an SNMP Group on page 84
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring SNMP Informs on page 101
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Copyright © 2012, Juniper Networks, Inc.

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;
               }
            }
```


Copyright © 2012, Juniper Networks, Inc.

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

Related Documentation

- Creating SNMPv3 Users on page 80
- Configuring MIB Views on page 66
- Defining Access Privileges for an SNMP Group on page 84
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring SNMP Informs on page 101
- Complete SNMPv3 Configuration Statements on page 76
- Example: SNMPv3 Configuration on page 108

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 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: SNMPv3 Configuration on page 108

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 {
```

 Copyright © 2012, Juniper Networks, Inc.

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

Related Documentation

- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Creating SNMPv3 Users Configuration
- Example: SNMPv3 Configuration on page 108

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 81
- Configuring SHA Authentication on page 81
- Configuring No Authentication on page 82

Configuring MD5 Authentication

To configure the message digest algorithm (MD5) as the authentication type for an SNMPv3 user, include the **authentication-md5** statement at the **[edit snmp v3 usm local-engine user *username*]** hierarchy level:

```
[edit snmp v3 usm local-engine user username]
authentication-md5 {
  authentication-password authentication-password;
}
```

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

SNMPv3 has special requirements when you create plain-text passwords on a router or switch:

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

Configuring SHA Authentication

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

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

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

SNMPv3 has special requirements when you create plain-text passwords on a router or switch:

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

Configuring No Authentication

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

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

Related Documentation

- Configuring the Encryption Type on page 82
- Defining Access Privileges for an SNMP Group on page 84
- Configuring the Access Privileges Granted to a Group on page 85
- Assigning Security Model and Security Name to a Group on page 89
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Configuring the Encryption Type

By default, encryption is set to none.

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

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

This topic includes the following sections:

- Configuring the Advanced Encryption Standard Algorithm on page 83
- Configuring the Data Encryption Algorithm on page 83
- Configuring Triple DES on page 83
- Configuring No Encryption on page 84

Copyright © 2012, Juniper Networks, Inc.

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 81
- Defining Access Privileges for an SNMP Group on page 84
- Configuring the Access Privileges Granted to a Group on page 85
- Assigning Security Model and Security Name to a Group on page 89
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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 66.

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

 Copyright © 2012, Juniper Networks, Inc.

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 81
- Configuring the Access Privileges Granted to a Group on page 85
- Assigning Security Model and Security Name to a Group on page 89
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Configuring the Access Privileges Granted to a Group

This topic includes the following sections:

- Configuring the Group on page 86
- Configuring the Security Model on page 86
- Configuring the Security Level on page 86
- Associating MIB Views with an SNMP User Group on page 87

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.

Copyright © 2012, Juniper Networks, Inc.

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 accessgroup 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 66.

This section describes the following topics related to this configuration:

- Configuring the Notify View on page 87
- Configuring the Read View on page 88
- Configuring the Write View on page 88

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 81
- Defining Access Privileges for an SNMP Group on page 84
- Assigning Security Model and Security Name to a Group on page 89
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Access Privilege Configuration on page 88

Example: Access Privilege Configuration

Define access privileges:

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


Copyright © 2012, Juniper Networks, Inc.

```
            }
          }
          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 85
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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 90
- Assigning Security Names to Groups on page 90
- Configuring the Group on page 90

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 80. For information about configuring a community string, see "Configuring the SNMPv3 Community" on page 106.

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

 Copyright © 2012, Juniper Networks, Inc.

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 84.

Example: Security Group Configuration

Assign security names to groups:

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

Related Documentation

- Assigning Security Model and Security Name to a Group on page 89
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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 101.

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.

Copyright © 2012, Juniper Networks, Inc.

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

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

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

Related Documentation

- Configuring the SNMPv3 Trap Notification on page 93
- Configuring the Trap Notification Filter on page 94
- Configuring the Trap Target Address on page 95
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Configuring the Remote Engine and Remote User on page 102
- Configuring the Inform Notification Type and Target Address on page 104
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Copyright © 2012, Juniper Networks, Inc.

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 96.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring the Trap Notification Filter on page 94
- Configuring the Trap Target Address on page 95
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Configuring SNMPv3 Trap Notification on page 93

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 91
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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 91
- Configuring the SNMPv3 Trap Notification on page 93

Copyright © 2012, Juniper Networks, Inc.

- Configuring the Trap Target Address on page 95
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Configuring the Trap Target Address

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

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

To specify where you want the traps to be sent and define what SNMPv1 and SNMPv2cc packets are allowed, include the **target-address** statement at the **[edit snmp v3]** hierarchy level:

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

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

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

```
[edit snmp v3 target-address target-address-name]
address address;
address-mask address-mask;
logical-system logical-system;
port port-number;
routing-instance instance;
tag-list tag-list;
target-parameters target-parameters-name;
```

This section includes the following topics:

- Configuring the Address on page 96
- Configuring the Address Mask on page 96
- Configuring the Port on page 96
- Configuring the Routing Instance on page 96
- Configuring the Trap Target Address on page 96
- Applying Target Parameters on page 97

Copyright © 2012, Juniper Networks, Inc.

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 106.

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.

 Copyright © 2012, Juniper Networks, Inc.

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 98.

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 93.

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 91
- Configuring the SNMPv3 Trap Notification on page 93
- Configuring the Trap Notification Filter on page 94
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Configuring the Tag List on page 98

Copyright © 2012, Juniper Networks, Inc.

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 91
- Configuring the Trap Target Address on page 95
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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.

Copyright © 2012, Juniper Networks, Inc.

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 99
- Configuring the Target Parameters on page 99

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 94.

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 100
- Configuring the Security Model on page 100
- Configuring the Security Level on page 100
- Configuring the Security Name on page 101

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.

Copyright © 2012, Juniper Networks, Inc.

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 91
- Configuring the SNMPv3 Trap Notification on page 93
- Configuring the Trap Notification Filter on page 94
- Configuring the Trap Target Address on page 95
- Configuring SNMP Informs on page 101
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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.

Copyright © 2012, Juniper Networks, Inc.

Informs are more reliable than traps, but they consume more network, router, and switch resources (see Figure 1 on page 102). 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 91.

Related Documentation

- Configuring SNMPv3 Traps on a Device Running Junos OS on page 91
- Configuring the Remote Engine and Remote User on page 102
- Configuring the Inform Notification Type and Target Address on page 104
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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 {
```

 Copyright © 2012, Juniper Networks, Inc.

```
            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 91
- Configuring SNMP Informs on page 101
- Configuring the Inform Notification Type and Target Address on page 104
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Configuring the Remote Engine ID and Remote Users on page 103

Example: Configuring the Remote Engine ID and Remote Users

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

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

```
            privacy-key "$9$MZZXxdwYgJUjlKJGiH5T69Au0IrlM7NbeK24
            aJDjO1IRylM8Xbwg1R24aJDjHqm5n/Ap0ORhn6evLXbwmf5T
            /CRhSyKM5QEcleW87-Vbs4JGD.mT-VwgaZkqfTznAphSrlM8yr
            Wx7dsYTzF36AtuO1EcpuNdwYoa69CuRhcyleM8rlaZGjq.O1IEhr";
        }
    }
}
```

Related Documentation

- Configuring the Remote Engine and Remote User on page 102
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Configuring the Inform Notification Type and Target Address

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

```
[edit snmp v3]
notify name {
    tag tag-name;
    type (trap | inform);
}
target-address target-address-name {
    address address;
    address-mask address-mask;
    logical-system logical-system;
    port port-number;
    retry-count number;
    routing-instance instance;
    tag-list tag-list;
    target-parameters target-parameters-name;
    timeout seconds;
}
target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
        message-processing-model (v1 | v2c | v3);
        security-level (authentication | none | privacy);
        security-model (usm | v1 | v2c);
        security-name security-name;
    }
}
```

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

tag *tag-name* defines the target addresses that are sent this notification. The notification is sent to all target addresses that have this tag in their tag list. The ***tag-name*** is not included in the notification. For information about how to configure the tag list, see "Configuring the Trap Target Address" on page 96.

type inform is the type of notification.

 Copyright © 2012, Juniper Networks, Inc.

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 91
- Configuring SNMP Informs on page 101
- Configuring the Remote Engine and Remote User on page 102
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: Configuring the Inform Notification Type and Target Address on page 105

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 104
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Configuring the SNMPv3 Community

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

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

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

community-index is the index for the SNMP community.

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

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

This section includes the following topics:

- Configuring the Community Name on page 106
- Configuring the Security Names on page 107
- Configuring the Tag on page 107

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

Copyright © 2012, Juniper Networks, Inc.

name define which MIB objects are available and the operations (read, write, or notify) allowed on those objects.

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

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

community-name is the community string for an SNMPv1 or SNMPv2c community.

If unconfigured, it is the same as the community index.

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

NOTE: Community names must be unique. You cannot configure the same community name at the **[edit snmp community]** and **[edit snmp v3 snmp-community *community-index*]** hierarchy levels. The configured community name at the **[edit snmp v3 snmp-community *community-index*]** hierarchy level is encrypted. You cannot view the community name after you have configured it and committed your changes. In the command-line interface (CLI), the community name is concealed.

Configuring the Security Names

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

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

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

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

Configuring the Tag

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

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

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

Related Documentation

- Creating SNMPv3 Users on page 80
- Complete SNMPv3 Configuration Statements on page 76

- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78
- Example: SNMPv3 Community Configuration on page 108

Example: SNMPv3 Community Configuration

Define an SNMP community:

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

Related Documentation

- Configuring the SNMPv3 Community on page 106
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

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;
```

 Copyright © 2012, Juniper Networks, Inc.

```
}
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;
            }
          }
        }
```

 Copyright © 2012, Juniper Networks, Inc.

```
        }
    }
    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 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 7

Standard SNMP Traps

- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Standard SNMP Version 1 Traps on page 113
- Standard SNMP Version 2 Traps on page 117
- Unsupported Standard SNMP Traps on page 121

Standard SNMP Traps Supported on Devices Running Junos OS

This topic provides pointers to the standard SNMP traps supported by the 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 Configuration Guide*.

- Standard SNMP Version 1 Traps on page 113
- Standard SNMP Version 2 Traps on page 117
- Standard SNMP Traps on EX Series Ethernet Switches on page 142
- Unsupported Standard SNMP Traps on page 121

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Standard SNMP Version 1 Traps

Table 6 on page 114 provides an overview of the standard traps for SNMPv1. The traps are organized first by trap category and then by trap name, and include their enterprise ID, generic trap number, and specific trap number. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. Traps that

do not have corresponding system logging severity levels are marked with an en dash (–) in the table.

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

Table 6: Standard Supported SNMP Version 1 Traps

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag	Supported On
Startup Notifications							
RFC 1215, *Conventions for Defining Traps for Use with the SNMP*	**authenticationFailure**	1.3.6.1.4.1.2636	4	0	Notice	SNMPD_ TRAP_ GEN_FAILURE	All devices running Junos OS.
	coldStart	1.3.6.1.4.1.2636	0	0	Critical	SNMPD_TRAP_ COLD_START	All devices running Junos OS.
	warmStart	1.3.6.1.4.1.2636	1	0	Error	SNMPD_TRAP_ WARM_START	All devices running Junos OS.
Link Notifications							
RFC 1215, *Conventions for Defining Traps for Use with the SNMP*	**linkDown**	1.3.6.1.4.1.2636	2	0	Warning	SNMP_ TRAP_ LINK_DOWN	All devices running Junos OS.
	linkUp	1.3.6.1.4.1.2636	3	0	Info	SNMP_TRAP_ LINK_UP	All devices running Junos OS.
Remote Operations Notifications							
RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*	**pingProbeFailed**	1.3.6.1.2.1.80.0	6	1	Info	SNMP_TRAP _PING_ PROBE_ FAILED	All devices running Junos OS.
	pingTestFailed	1.3.6.1.2.1.80.0	6	2	Info	SNMP_TRAP_ PING_TEST _FAILED	All devices running Junos OS.
	pingTestCompleted	1.3.6.1.2.1.80.0	6	3	Info	SNMP_TRAP_ PING_TEST_ COMPLETED	All devices running Junos OS.
	traceRoutePathChange	1.3.6.1.2.1.81.0	6	1	Info	SNMP_TRAP_ TRACE_ROUTE_ PATH_CHANGE	All devices running Junos OS.
	traceRouteTestFailed	1.3.6.1.2.1.81.0	6	2	Info	SNMP_TRAP_ TRACE_ROUTE_ TEST_FAILED	All devices running Junos OS.
	traceRouteTestCompleted	1.3.6.1.2.1.81.0	6	3	Info	SNMP_TRAP_ TRACE_ROUTE_ TEST_COMPLETED	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 6: Standard Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag	Supported On
RMON Alarms							
RFC 2819a, *RMON MIB*	**fallingAlarm**	1.3.6.1.2.1.16	6	2	–	–	All devices running Junos OS.
	risingAlarm	1.3.6.1.2.1.16	6	1	–	–	All devices running Junos OS.
Routing Notifications							
BGP 4 MIB	**bgpEstablished**	1.3.6.1.2.1.15.7	6	1	–	–	M, T, MX, J, EX, and SRX for branch devices.
	bgpBackwardTransition	1.3.6.1.2.1.15.7	6	2	–	–	M, T, MX, J, EX, and SRX for branch devices.

Copyright © 2012, Juniper Networks, Inc.

Table 6: Standard Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag	Supported On
OSPF TRAP MIB	**ospfVirtIfStateChange**	1.3.6.1.2.1.14.16.2	6	1	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfNbrStateChange	1.3.6.1.2.1.14.16.2	6	2	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2	6	3	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfIfConfigError	1.3.6.1.2.1.14.16.2	6	4	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2	6	5	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2	6	6	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfVirtIfAuthFailure	1.3.6.1.2.1.14.16.2	6	7	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	8	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	9	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfTxRetransmit	1.3.6.1.2.1.14.16.2	6	10	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2	6	11	–	–	M, T, MX, J, EX, and SRX for branch devices.
	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2	6	13	–	–	M, T, MX, J, EX, and SRX for branch devices.

Copyright © 2012, Juniper Networks, Inc.

Table 6: Standard Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag	Supported On
	ospfIfStateChange	1.3.6.1.2.1.14.16.2	6	16	–	–	M, T, MX, J, EX, and SRX for branch devices.
VRRP Notifications							
RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*	**vrrpTrapNewMaster**	1.3.6.1.2.1.68	6	1	Warning	VRRPD_NEW MASTER_TRAP	All devices running Junos OS.
	vrrpTrapAuthFailure	1.3.6.1.2.1.68	6	2	Warning	VRRPD_AUTH_ FAILURE_TRAP	All devices running Junos OS.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Standard SNMP Version 2 Traps

Table 7 on page 118 provides an overview of the standard SNMPv2 traps supported by the Junos OS. The traps are organized first by trap category and then by trap name and include their **snmpTrapOID**. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. Traps that do not have corresponding system logging severity levels are marked with an en dash (–) in the table.

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

Table 7: Standard Supported SNMP Version 2 Traps

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
Startup Notifications					
RFC 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*	**coldStart**	1.3.6.1.6.3.1.1.5.1	Critical	SNMPD_TRAP_COLD_START	All devices running Junos OS.
	warmStart	1.3.6.1.6.3.1.1.5.2	Error	SNMPD_TRAP_WARM_START	All devices running Junos OS.
	authenticationFailure	1.3.6.1.6.3.1.1.5.5	Notice	SNMPD_TRAP_GEN_FAILURE	All devices running Junos OS.
Link Notifications					
RFC 2863, *The Interfaces Group MIB*	**linkDown**	1.3.6.1.6.3.1.1.5.3	Warning	SNMP_TRAP_LINK_DOWN	All devices running Junos OS.
	linkUp	1.3.6.1.6.3.1.1.5.4	Info	SNMP_TRAP_LINK_UP	All devices running Junos OS.
Remote Operations Notifications					
RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*	**pingProbeFailed**	1.3.6.1.2.1.80.0.1	Info	SNMP_TRAP_PING_PROBE_FAILED	All devices running Junos OS.
	pingTestFailed	1.3.6.1.2.1.80.0.2	Info	SNMP_TRAP_PING_TEST_FAILED	All devices running Junos OS.
	pingTestCompleted	1.3.6.1.2.1.80.0.3	Info	SNMP_TRAP_PING_TEST_COMPLETED	All devices running Junos OS.
	traceRoutePathChange	1.3.6.1.2.1.81.0.1	Info	SNMP_TRAP_TRACE_ROUTE_PATH_CHANGE	All devices running Junos OS.
	traceRouteTestFailed	1.3.6.1.2.1.81.0.2	Info	SNMP_TRAP_TRACE_ROUTE_TEST_FAILED	All devices running Junos OS.
	traceRouteTestCompleted	1.3.6.1.2.1.81.0.3	Info	SNMP_TRAP_TRACE_ROUTE_TEST_COMPLETED	All devices running Junos OS.
RMON Alarms					
RFC 2819a, *RMON MIB*	**fallingAlarm**	1.3.6.1.2.1.16.0.1	–	–	All devices running Junos OS.
	risingAlarm	1.3.6.1.2.1.16.0.2	–	–	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 7: Standard Supported SNMP Version 2 Traps *(continued)*

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
Routing Notifications					
BGP 4 MIB	**bgpEstablished**	1.3.6.1.2.1.15.7.1	–	–	M, MX, T, EX, J, and SRX for branch devices.
	bgpBackwardTransition	1.3.6.1.2.1.15.7.2	–	–	M, MX, T, EX, J, and SRX for branch devices.
OSPF Trap MIB	**ospfVirtIfStateChange**	1.3.6.1.2.1.14.16.2.1	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfNbrStateChange	1.3.6.1.2.1.14.16.2.2	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2.3	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfIfConfigError	1.3.6.1.2.1.14.16.2.4	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.5	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2.6	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfVirtIfAuthFailure	1.3.6.1.2.1.14.16.2.7	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2.8	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2.9	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfTxRetransmit	1.3.6.1.2.1.14.16.2.10	–	–	M, MX, T, EX, J, and SRX for branch devices.

Copyright © 2012, Juniper Networks, Inc.

Table 7: Standard Supported SNMP Version 2 Traps *(continued)*

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2.11	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2.13	–	–	M, MX, T, EX, J, and SRX for branch devices.
	ospfIfStateChange	1.3.6.1.2.1.14.16.2.16	–	–	M, MX, T, EX, J, and SRX for branch devices.
RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*	**vrrpTrapNewMaster**	1.3.6.1.2.1.68.0.1	Warning	VRRPD_ NEWMASTER_TRAP	All devices running Junos OS.
	vrrpTrapAuthFailure	1.3.6.1.2.1.68.0.2	Warning	VRRPD_AUTH_ FAILURE_ TRAP	All devices running Junos OS.

The Junos OS also supports the following standard SNMP version 2 traps:

• SNMP Version 2 MPLS Traps on page 120

SNMP Version 2 MPLS Traps

The Junos OS supports the MPLS SNMP version 2 traps defined in RFC 3812, *Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base*.

You can disable the MPLS traps by including the **no-trap** option at the **[edit protocol mpls log-updown]** hierarchy level. For information about disabling the generation of MPLS traps, see the *Junos OS MPLS Applications Configuration Guide*.

The Junos OS supports the following MPLS traps:

- **mplsTunnelUp**—Generated when an **mplsTunnelOperStatus** object for one of the configured tunnels leaves the **down** state and transitions into another state, other than the **notPresent** state.
- **mplsTunnelDown**—Generated when an **mplsTunnelOperStatus** object for one of the configured tunnels enters the **down** state from a state other than the **notPresent** state.

NOTE: When an LSP flaps, only the ingress and egress routers of that LSP generate the **mplsTunnelUp** and **mplsTunnelDown** traps. Previously, all the routers associated with an LSP—that is, the ingress, egress, and transit routers—used to generate the traps when the LSP flaps.

Copyright © 2012, Juniper Networks, Inc.

- **mplsTunnelRerouted**—Generated when a tunnel is rerouted.
- **mplsTunnelReoptimized**—Generated when a tunnel is reoptimized.

NOTE: In Junos OS Release 8.3 and earlier, **mplsTunnelReoptimized** was generated every time the optimization timer expired; that is, when the optimization timer exceeded the value set for the **optimize-timer** statement at the **[edit protocols mpls label-switched-path path-name]** hierarchy level. However, in Release 8.4 and later, this trap is generated only when the path is reoptimized, and not when the optimization timer expires.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Unsupported Standard SNMP Traps

Standard SNMP traps that are defined in MIBs supported by the Junos OS but are not generated by the Junos OS are shown in Table 8 on page 122.

Table 8: Unsupported Standard SNMP Traps

MIB	Trap Name	Description
isismib.mib	isisDatabaseOverload	Generated when the system enters or leaves the overload state.
	isisManualAddressDrops	Generated when one of the manual **areaAddresses** assigned to the system is ignored when computing routes.
	isisCorruptedLSPDetected	Generated when an LSP stored in memory becomes corrupted.
	isisAttemptToExceedMaxSequence	Generated when the sequence number on a generated LSP wraps the 32-bit sequence counter and the number is purged.
	isisIDLenMismatch	Generated when a protocol data unit (PDU) is received with a different value for the system ID length. This trap includes an index to identify the circuit where the PDU was received and the PDU header.
	isisMaxAreaAddressesMismatch	Generated when a PDU with a different value for the maximum area addresses is received.
	isisOwnLSPPurge	Generated when a PDU is received with a system ID and zero age. This notification includes the circuit index if available.
	isisSequenceNumberSkip	Generated when an LSP is received with a system ID and different contents, indicating the LSP might require a higher sequence number.
	isisAuthenticationTypeFailure	Generated when a PDU with the wrong authentication type field is received.
	isisAuthenticationFailure	Generated when a PDU with an incorrect authentication information field is received.
	isisVersionSkew	Generated when a hello PDU from an IS running a different version of the protocol is received.
	isisAreaMismatch	Generated when a hello PDU from an IS which does not share any area address is received.
	isisRejectedAdjacency	Generated when a hello PDU from an IS is received, but no adjacency is established because of a lack of resources.
	isisLSPTooLargeToPropagate	Generated when a link-state PDU that is larger than the **dataLinkBlockSize** for a circuit is attempted, but not propagated.
	isisOriginatingLSPBufferSizeMismatch	

Copyright © 2012, Juniper Networks, Inc.

Table 8: Unsupported Standard SNMP Traps *(continued)*

MIB	Trap Name	Description
		Generated when a Level 1 link-state PDU or Level 2 link-state PDU is received that is larger than the local value for originating **L1LSPBufferSize** or originating **L2LSPBufferSize**, respectively, or when a Level 1 link-state PDU or Level 2 link-state PDU is received containing the originating **LSPBufferSize** option and the value in the PDU option field does not match the local value for originating **L1LSPBufferSize** or originating **L2LSPBufferSize**, respectively.
	isisProtocolsSupportedMismatch	Generated when a nonpseudonode, segment 0 link-state PDU is received that has no matching protocols.
l3vpnmib.mib	**mplsVrfIfUp**	Generated when the **ifOperStatus** of an interface associated with a VRF table changes to the **up(1)** state, or when an interface with **ifOperStatus = up(1)** is associated with a VRF table.
	mplsVrfIfDown	Generated when the **ifOperStatus** of an interface associated with a VRF table changes to the **down(1)** state, or when an interface with **ifOperStatus = up(1)** state is disassociated from a VRF table.
	mplsNumVrfRouteMidThreshExceeded	Generated when the number of routes contained by the specified VRF table exceeds the value indicated by **mplsVrfMidRouteThreshold**.
	mplsNumVrfRouteMaxThreshExceeded	Generated when the number of routes contained by the specified VRF table reaches or attempts to exceed the maximum allowed value as indicated by **mplsVrfMaxRouteThreshold**.
	mplsNumVrfSecIllglLblThrshExcd	Generated when the number of illegal label violations on a VRF table as indicated by **mplsVpnVrfSecIllegalLblVltns** has exceeded **mplsVpnVrfSecIllegalLblRcvThrsh**.
msdpmib.mib	**msdpEstablished**	Generated when the Multicast Source Discovery Protocol (MSDP) finite state machine (FSM) enters the **Established** state.
	msdpBackwardTransition	Generated when the MSDP FSM moves from a higher numbered state to a lower numbered state.
ospf2trap.mib	**ospfOriginateLsa**	Generated when a new LSA is originated by the router because of a topology change.
	ospfLsdbOverflow	Generated when the number of LSAs in the router's link-state database exceeds the value of **ospfExtLsdbLimit**.
	ospfLsdbApproachingOverflow	Generated when the number of LSAs in the router's link-state database exceeds 90% of the value of **ospfExtLsdbLimit**.

Copyright © 2012, Juniper Networks, Inc.

Table 8: Unsupported Standard SNMP Traps *(continued)*

MIB	Trap Name	Description
rfc1747.mib	**sdlcPortStatusChange**	Generated when the state of an SDLC port transitions to active or inactive.
	sdlcLSStatusChange	Generated when the state of an SDLC link station transitions to contacted or disconnected.
rfc2115a.mib	**frDLCIStatusChange**	Generated when a virtual circuit changes state (has been created or invalidated, or has toggled between the active and inactive states).
rfc2662.mib	**adslAtucRateChangeTrap**	Generated when the ATUCs transmit rate has changed (RADSL mode only).
	adslAtucPerfLofsThreshTrap	Generated when the loss of framing 15-minute interval threshold is reached.
	adslAtucInitFailureTrap	Generated when ATUC initialization fails.
	adslAturPerfLprsThreshTrap	Generated when the loss of power 15-minute interval threshold is reached.
	adslAturRateChangeTrap	Generated when the ATURs transmit rate changes (RADSL mode only).
rfc3020.mib	**mfrMibTrapBundleLinkMismatch**	Generated when a bundle link mismatch is detected.
rfc3813.mib	**mplsXCUp**	Generated when **mplsXCOperStatus** for one or more contiguous entries in **mplsXCTable** enters the **up(1)** state from some other state.
	mplsXCDown	Generated when **mplsXCOperStatus** for one or more contiguous entries in **mplsXCTable** enters the **down(2)** state from some other state.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 8

Juniper Networks Enterprise-Specific SNMP Traps

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Juniper Networks Enterprise-Specific SNMP Version 1 Traps on page 126
- Juniper Networks Enterprise-Specific SNMP Version 2 Traps on page 133
- Juniper Networks Enterprise-Specific BGP Traps on page 140
- Juniper Networks Enterprise-Specific LDP Traps on page 141
- Juniper Networks Enterprise-Specific Traps on EX Series Switches on page 141
- Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers on page 142
- Standard SNMP Traps on EX Series Ethernet Switches on page 142
- Configuring SNMPv3 Traps on a Device Running Junos OS on page 143

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 on page 126
- Juniper Networks Enterprise-Specific SNMP Version 2 Traps on page 133
- Juniper Networks Enterprise-Specific BGP Traps on page 140
- Juniper Networks Enterprise-Specific LDP Traps on page 141
- Juniper Networks Enterprise-Specific License MIB Notifications on page 435
- Juniper Networks Enterprise-Specific MIMSTP Traps
- Juniper Networks Enterprise-Specific MPLS Traps

Copyright © 2012, Juniper Networks, Inc.

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 Configuration Guide*.

- Juniper Networks Enterprise-Specific Traps on EX Series Switches on page 141
- Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers on page 142

Related Documentation

- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Juniper Networks Enterprise-Specific SNMP Version 1 Traps

The Junos OS supports enterprise-specific SNMP version 1 traps shown in Table 9 on page 126. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. Traps that do not have corresponding system logging severity levels are marked with an en dash (–).

For more information about system log messages, see the *Junos OS System Log Messages Reference*. For more information about configuring system logging, see the *Junos OS System Basics Configuration Guide*. To view the Juniper Networks enterprise-specific SNMP version 2 traps, see "Juniper Networks Enterprise-Specific SNMP Version 2 Traps" on page 133. For more information about chassis traps, see "Chassis Traps" on page 355.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
Chassis Notifications (Alarm Conditions)							
Chassis MIB (**jnx-chassis.mib**)	**jnxPowerSupplyFailure**	1.3.6.1.4.1.2636.4.1	6	1	Warning	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3	Alert	CHASSISD_SNMP_TRAP	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1	6	4	Critical	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9	Warning	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11	Notice	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12	Warning	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFEBSwitchover	1.3.6.1.4.1.2636.4.1	6	13	Warning	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxHardDiskFailed	1.3.6.1.4.1.2636.4.1	6	14	Warning	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxHardDiskMissing	1.3.6.1.4.1.2636.4.1	6	15	Warning	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
	jnxPowerSupplyOk	1.3.6.1.4.1.2636.4.2	6	1	Critical	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2	Critical	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3	Alert	CHASSISD_ SNMP_ TRAP	All devices running Junos OS.
Configuration Notifications							
Configuration Management MIB (**jnx-configmgmt.mib**)	**jnxCmCfgChange**	1.3.6.1.4.1.2636.4.5	6	1	–	–	All devices running Junos OS.
	jnxCmRescueChange	1.3.6.1.4.1.2636.4.5	6	2	–	–	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
Link Notifications							
Flow Collection Services MIB (**jnx-coll.mib**)	**jnxCollUnavailableDest**	1.3.6.1.4.1.2636.4.8	6	1	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollUnavailableDestCleared	1.3.6.1.4.1.2636.4.8	6	2	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollUnsuccessfulTransfer	1.3.6.1.4.1.2636.4.8	6	3	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFlowOverload	1.3.6.1.4.1.2636.4.8	6	4	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFlowOverloadCleared	1.3.6.1.4.1.2636.4.8	6	5	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollMemoryUnavailable	1.3.6.1.4.1.2636.4.8	6	6	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollMemoryAvailable	1.3.6.1.4.1.2636.4.8	6	7	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFtpSwitchover	1.3.6.1.4.1.2636.4.8	6	8	–	–	Devices that run Junos OS and have collector PICs installed.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
Passive Monitoring MIB (jnx-pmon.mib)	jnxPMonOverloadSet	1.3.6.1.4.1.2636.4.7.0.1	6	1	–	–	Devices that run Junos OS and have PICs that support passive monitoring installed.
	jnxPMonOverloadCleared	1.3.6.1.4.1.2636.4.7.0.2	6	2	–	–	Devices that run Junos OS and have PICs that support passive monitoring installed.
SONET APS MIB (jnx-sonetaps.mib)	apsEventChannelMismatch	1.3.6.1.4.1.2636.3.24.2	6	3	–	–	Devices that run Junos OS and have SONET PICs installed.
	apsEventPSBF	1.3.6.1.4.1.2636.3.24.2	6	4	–	–	Devices that run Junos OS and have SONET PICs installed.
	apsEventFEPLF	1.3.6.1.4.1.2636.3.24.2	6	5	–	–	Devices that run Junos OS and have SONET PICs installed.
Remote Operations							
PING MIB (jnx-ping.mib)	jnxPingRttThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	1	–	–	All devices running Junos OS.
	jnxPingRttStdDevThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	2	–	–	All devices running Junos OS.
	jnxPingRttJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	3	–	–	All devices running Junos OS.
	jnxPingEgressThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	4	–	–	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
	jnxPingEgressStdDev ThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	5	–	–	All devices running Junos OS.
	jnxPingEgressJitter ThresholdExceeded	1.3.6.1.4.1.2636.4.9	6	6	–	–	All devices running Junos OS.
	jnxPingIngressThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	7	–	–	All devices running Junos OS.
	jnxPingIngressStddevThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	8	–	–	All devices running Junos OS.
	jnxPingIngressJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9	6	9	–	–	All devices running Junos OS.
Routing Notifications							
BFD Experimental MIB (**jnx-bfd-exp.mib**)	**bfdSessUp**	1.3.6.1.4.1. 2636.5.3.1	6	1	–	–	All devices running Junos OS.
	bfdSessDown	1.3.6.1.4.1. 2636.5.3.1	6	2	–	–	All devices running Junos OS.
LDP MIB (**jnx-ldp.mib**)	**jnxLdpLspUp**	1.3.6.1.4.1.2636.4.4	6	1	–	–	M, T, and MX Series routers.
	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4	6	2	–	–	M, T, and MX Series routers.
	jnxLdpSesUp	1.3.6.1.4.1.2636.4.4	6	3	–	–	M, T, and MX Series routers.
	jnxLdpSesDown	1.3.6.1.4.1.2636.4.4	6	4	–	–	M, T, and MX Series routers.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps *(continued)*

Defined in	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag	Supported On
MPLS MIB (jnx-mpls.mib)	**mplsLspUp** (Deprecated)	1.3.6.1.4.1.2636.3.2.4	6	1	–	–	
	mplsLspDown (Deprecated)	1.3.6.1.4.1.2636.3.2.4	6	2	–	–	
	mplsLspChange (Deprecated)	1.3.6.1.4.1.2636.3.2.4	6	3	–	–	
	mplsLspPathDown (Deprecated)	1.3.6.1.4.1.2636.3.2.4	6	4	–	–	
VPN MIB (jnx-vpn.mib)	**jnxVpnIfUp**	1.3.6.1.4.1.2636.3.26	6	1	–	–	M, T, and MX Series routers.
	jnxVpnIfDown	1.3.6.1.4.1.2636.3.26	6	2	–	–	M, T, and MX Series routers.
	jnxVpnPwUp	1.3.6.1.4.1.2636.3.26	6	3	–	–	M, T, and MX Series routers.
	jnxVpnPwDown	1.3.6.1.4.1.2636.3.26	6	4	–	–	M, T, and MX Series routers.
RMON Alarms							
RMON MIB (jnx-rmon.mib)	**jnxRmonAlarmGetFailure**	1.3.6.1.4.1.2636.4.3	6	1	–	–	All devices running Junos OS.
	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3	6	2	–	–	All devices running Junos OS.
SONET Alarms							
SONET MIB (jnx-sonet.mib)	**jnxSonetAlarmSet**	1.3.6.1.4.1.2636.4.6	6	1	–	–	Devices that run Junos OS and have SONET PICs installed.
	jnxSonetAlarmCleared	1.3.6.1.4.1.2636.4.6	6	2	–	–	Devices that run Junos OS and have SONET PICs installed.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31

Copyright © 2012, Juniper Networks, Inc.

- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Juniper Networks Enterprise-Specific SNMP Version 2 Traps

The Junos OS supports the enterprise-specific SNMP version 2 traps shown in Table 10 on page 133. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. Traps that do not have corresponding system logging severity levels are marked with an en dash (–).

For more information about system messages, see the *Junos OS System Log Messages Reference*. For more information about configuring system logging, see the *Junos OS System Basics Configuration Guide*. To view the Juniper Networks enterprise-specific SNMP version 1 traps, see "Juniper Networks Enterprise-Specific SNMP Version 1 Traps" on page 126. For more information about chassis traps, see "Chassis Traps" on page 355.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
Chassis (Alarm Conditions) Notifications					
Chassis MIB (**jnx-chassis.mib**)	**jnxPowerSupplyFailure**	1.3.6.1.4.1.2636.4.1.1	Alert	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1.4	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
	jnxFruFailed	1.3.6.1.4.1.2636.4.1.9	Warning	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruOffline	1.3.6.1.4.1.2636.4.1.10	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruOnline	1.3.6.1.4.1.2636.4.1.11	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFruCheck	1.3.6.1.4.1.2636.4.1.12	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFEBSwitchover	1.3.6.1.4.1.2636.4.1.13	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxHardDiskFailed	1.3.6.1.4.1.2636.4.1.14	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxHardDiskMissing	1.3.6.1.4.1.2636.4.1.15	Notice	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxFanOK	1.3.6.1.4.1.2636.4.2.2	Critical	CHASSISD_SNMP_TRAP	All devices running Junos OS.
	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	Alert	CHASSISD_SNMP_TRAP	All devices running Junos OS.
Configuration Notifications					
Configuration Management MIB (**jnx-cfgmgmt.mib**)	**jnxCmCfgChange**	1.3.6.1.4.1.2636.4.5.0.1	–	–	All devices running Junos OS.
	jnxCmRescueChange	1.3.6.1.4.1.2636.4.5.0.2	–	–	All devices running Junos OS.
Link Notifications					

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
Flow Collection Services MIB (jnx-coll.mib)	**jnxCollUnavailableDest**	1.3.6.1.4.1.2636.4.8.0.1	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollUnavailableDestCleared	1.3.6.1.4.1.2636.4.8.0.2	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollUnsuccessfulTransfer	1.3.6.1.4.1.2636.4.8.0.3	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFlowOverload	1.3.6.1.4.1.2636.4.8.0.4	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFlowOverloadCleared	1.3.6.1.4.1.2636.4.8.0.5	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollMemoryUnavailable	1.3.6.1.4.1.2636.4.8.0.6	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollMemoryAvailable	1.3.6.1.4.1.2636.4.8.0.7	–	–	Devices that run Junos OS and have collector PICs installed.
	jnxCollFtpSwitchover	1.3.6.1.4.1.2636.4.8.0.8	–	–	Devices that run Junos OS and have collector PICs installed.
PMON MIB (jnx-pmon.mib)	**jnxPMonOverloadSet**	1.3.6.1.4.1.2636.4.7.0.1	–	–	Devices that run Junos OS and have PICs that support passive monitoring installed.
	jnxPMonOverloadCleared	1.3.6.1.4.1.2636.4.7.0.2	–	–	Devices that run Junos OS and have PICs that support passive monitoring installed.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
SONET APS MIB (**jnx-sonetaps.mib**)	**apsEventChannelMismatch**	1.3.6.1.4.1.2636.3.24.2.0.3	–	–	Devices that run Junos OS and have SONET PICs installed.
	apsEventPSBF	1.3.6.1.4.1.2636.3.24.2.0.4	–	–	Devices that run Junos OS and have SONET PICs installed.
	apsEventFEPLF	1.3.6.1.4.1.2636.3.24.2.0.5	–	–	Devices that run Junos OS and have SONET PICs installed.
Remote Operations Notifications					
PING MIB (**jnx-ping.mib**)	**jnxPingRttThreshold Exceeded**	1.3.6.1.4.1.2636.4.9.0.1	–	–	All devices running Junos OS.
	jnxPingRttStdDevThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.2	–	–	All devices running Junos OS.
	jnxPingRttJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.3	–	–	All devices running Junos OS.
	jnxPingEgressThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.4	–	–	All devices running Junos OS.
	jnxPingEgressStdDevThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.5	–	–	All devices running Junos OS.
	jnxPingEgressJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.6	–	–	All devices running Junos OS.
	jnxPingIngressThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.7	–	–	All devices running Junos OS.
	jnxPingIngressStddevThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.8	–	–	All devices running Junos OS.
	jnxPingIngressJitterThreshold Exceeded	1.3.6.1.4.1.2636.4.9.0.9	–	–	All devices running Junos OS.
Routing Notifications					
BFD Experimental MIB (**jnx-bfd-exp.mib**)	**bfdSessUp**	1.3.6.1.4.1.2636.5.3.1.0.1	–	–	All devices running Junos OS.
	bfdSessDown	1.3.6.1.4.1.2636.5.3.1.0.2	–	–	All devices running Junos OS.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
BGP4 V2 MIB (jnx-bgpmib2.mib)	jnxBgpM2Established	1.3.6.1.4.1.2636.5.1.1.1.0.1	–	–	All devices running Junos OS.
	jnxBgpM2BackwardTransition	1.3.6.1.4.1.2636.5.1.1.1.0.2	–	–	All devices running Junos OS.
DHCP MIB (jnx-dhcp.mib)	jnxJdhcpLocalServer DuplicateClient	1.3.6.1.4.1.2636.3.61.61.1.3.1	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer InterfaceLimitExceeded	1.3.6.1.4.1.2636.3.61.61.1.3.2	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer InterfaceLimitAbated	1.3.6.1.4.1.2636.3.61.61.1.3.3	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer Health	1.3.6.1.4.1.2636.3.61.61.1.3.4	–	–	All devices running Junos OS.
	jnxJdhcpRelayInterface LimitExceeded	1.3.6.1.4.1.2636.3.61.61.2.3.1	–	–	All devices running Junos OS.
	jnxJdhcpRelayInterface LimitAbated	1.3.6.1.4.1.2636.3.61.61.2.3.2	–	–	All devices running Junos OS.
DHCPv6MIB (jnx-dhcpv6.mib)	jnxJdhcpv6LocalServer InterfaceLimitExceeded	1.3.6.1.4.1.2636.3.62.62.2.3.1	–	–	All devices running Junos OS.
	jnxJdhcpv6LocalServer InterfaceLimitAbated	1.3.6.1.4.1.2636.3.62.62.2.3.2	–	–	All devices running Junos OS.
	jnxJdhcpv6LocalServer Health	1.3.6.1.4.1.2636.3.62.62.2.3.3	–	–	All devices running Junos OS.
LDP MIB (jnx-ldp.mib)	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4.0.1	–	–	M, T, and MX Series routers.
	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4.0.2	–	–	M, T, and MX Series routers.
	jnxLdpSesUp	1.3.6.1.4.1.2636.4.4.0.3	–	–	M, T, and MX Series routers.
	jnxLdpSesDown	1.3.6.1.4.1.2636.4.4.0.4	–	–	M, T, and MX Series routers.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
MPLS MIB (**jnx-mpls.mib**)	**mplsLspUp** (Deprecated)	1.3.6.1.4.1.2636.3.2.4.1	–	–	
	mplsLspInfoUp	1.3.6.1.4.1.2636.3.2.0.1	–	–	M, T, and MX Series routers.
	mplsLspDown (Deprecated)	1.3.6.1.4.1.2636.3.2.4.2	–	–	
	mplsLspInfoDown	1.3.6.1.4.1.2636.3.2.0.2	–	–	M, T, and MX Series routers.
	mplsLspChange (Deprecated)	1.3.6.1.4.1.2636.3.2.4.3	–	–	
	mplsLspInfoChange	1.3.6.1.4.1.2636.3.2.0.3	–	–	M, T, and MX Series routers.
	mplsLspPathDown (Deprecated)	1.3.6.1.4.1.2636.3.2.4.4	–	–	
	mplsLspInfoPathDown	1.3.6.1.4.1.2636.3.2.0.4	–	–	M, T, and MX Series routers.
	mplsLspInfoPathUp	1.3.6.1.4.1.2636.3.2.0.5	–	–	M, T, and MX Series routers.
VPN MIB (**jnx-vpn.mib**)	**jnxVpnIfUp**	1.3.6.1.4.1.2636.3.26.0.1	–	–	M, T, and MX Series routers.
	jnxVpnIfDown	1.3.6.1.4.1.2636.3.26.0.2	–	–	M, T, and MX Series routers.
	jnxVpnPwUp	1.3.6.1.4.1.2636.3.26.0.3	–	–	M, T, and MX Series routers.
	jnxVpnPwDown	1.3.6.1.4.1.2636.3.26.0.4	–	–	M, T, and MX Series routers.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
AAA MIB (jnx-user-aaa.mib)	jnxAccessAuthAddressPoolHighThreshold	1.3.6.1.4.1.2636.3.51.1.0.5	–	–	J Series and SRX Series devices.
	jnxAccessAuthAddressPoolAbateThreshold	1.3.6.1.4.1.2636.3.51.1.0.6	–	–	J Series and SRX Series devices.
	jnxAccessAuthAddressPoolOutOfAddresses	1.3.6.1.4.1.2636.3.51.1.0.7	–	–	J Series and SRX Series devices.
	jnxAccessAuthAddressPoolOutOfMemory	1.3.6.1.4.1.2636.3.51.1.0.8	–	–	J Series and SRX Series devices.
	jnxAccessAuthService Up	1.3.6.1.4.1.2636.3.51.1.0.1	–	–	J Series and SRX Series devices.
	jnxAccessAuthService Down	1.3.6.1.4.1.2636.3.51.1.0.2	–	–	J Series and SRX Series devices.
	jnxAccessAuthServer Disabled	1.3.6.1.4.1.2636.3.51.1.0.3	–	–	J Series and SRX Series devices.
	jnxAccessAuthServer Enabled	1.3.6.1.4.1.2636.3.51.1.0.4	–	–	J Series and SRX Series devices.
	jnxJsFwAuthFailure	1.3.6.1.4.1.2636.3.39.1.2.1.0.1	–	–	J Series and SRX Series devices.
Access Authentication Methods MIB (jnx-js-auth.mib)	jnxJsFwAuthServiceUp	1.3.6.1.4.1.2636.3.39.1.2.1.0.2	–	–	J Series and SRX Series devices.
	jnxJsFwAuthServiceDown	1.3.6.1.4.1.2636.3.39.1.2.1.0.3	–	–	J Series and SRX Series devices.
	jnxJsFwAuthCapacityExceeded	1.3.6.1.4.1.2636.3.39.1.2.1.0.4	–	–	J Series and SRX Series devices.
	jnxJsNatAddrPoolThresholdStatus	1.3.6.1.4.1.2636.3.39.1.7.1.0.1	–	–	J Series and SRX Series devices.
Network Address Translation Resources-Monitoring MIB (jnxNatMIB)	jnxNatAddrPoolUtil	1.3.6.1.4.1.2636.3.59.1.2.1	–	–	M Series and MX Series routers
	jnxNatTrapSrcPoolName	1.3.6.1.4.1.2636.3.59.1.2.2	–	–	M Series and MX Series routers
	jnxNatAddrPoolThresholdStatus	1.3.6.1.4.1.2636.3.59.1.0.1	–	–	M Series and MX Series routers

Copyright © 2012, Juniper Networks, Inc.

Table 10: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps *(continued)*

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
Network Address Translation MIB (jnx-js-nat.mib)	**jnxJsScreen Attack**	1.3.6.1.4.1.2636.3.39.1.8.1.0.1	Warning	RT_SCREEN_ICMP, RT_SCREEN_IP, RT_SCREEN_SESSION_LIMIT, RT_SCREEN_TCP, RT_SCREEN_UDP	J Series and SRX Series devices.
Security Screening Objects MIB (jnx-js-screening.mib)	**jnxJsScreenCfg Change**	1.3.6.1.4.1.2636.3.39.1.8.1.0.2	–	–	J Series and SRX Series devices.
RMON Alarms					
RMON MIB (jnx-rmon.mib)	**jnxRmonGetOk**	1.3.6.1.4.1.2636.4.3.0.2	–	–	All devices running Junos OS.
SONET Alarms					
SONET MIB (jnx-sonet.mib)	**jnxSonetAlarm Cleared**	1.3.6.1.4.1.2636.4.6.0.2	–	–	Devices that run Junos OS and have SONET PICs installed.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Juniper Networks Enterprise-Specific BGP Traps

Junos OS Release 10.0 introduced two enterprise-specific traps, **jnxBgpM2Established** and **jnxBgpM2BackwardTransition**, to support BGP clients that follow IPv6 addressing. The previous versions of the Junos OS supported only the standard BGP traps, **BgpM2Established** and **BgpM2BackwardTransition**. The standard BGP traps support only IPv4 addresses, and return 0.0.0.0 as the IP address of the BGP remote peer if the remote peer is configured with an IPv6 address. The newly introduced enterprise-specific traps support IPv6 addressing and contain the proper IPv6 address of the remote peer if the remote peer is configured with an IPv6 address. However, the Junos OS continues to

 Copyright © 2012, Juniper Networks, Inc.

support the standard traps and generates them when the BGP clients are configured with IPv4 addresses.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Juniper Networks Enterprise-Specific LDP Traps

For information about the enterprise-specific LDP traps, see LDP MIB.

Disabling LDP Traps

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

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Juniper Networks Enterprise-Specific Traps on EX Series Switches

EX Series switches support the following enterprise-specific traps:

- **jnxVccpPortUp**
- **jnxVccpPortDown**
- **jnxBootFromBackup**

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Copyright © 2012, Juniper Networks, Inc.

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

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

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

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

For information about Juniper Networks enterprise-specific MIMSTP traps that are supported on MX Series routers, see Juniper Networks Enterprise-Specific MIMSTP Traps.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Standard SNMP Traps on EX Series Ethernet Switches

Apart from the standard SNMP traps listed in the preceding sections, the Junos OS also supports the following standard traps on the EX Series Ethernet Switches:

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

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

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 101.

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

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

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

```
[edit snmp v3]
notify name {
   tag tag-name;
   type trap;
}
notify-filter name {
   oid object-identifier (include | exclude);
}
target-address target-address-name {
   address address;
   address-mask address-mask;
   logical-system logical-system;
   port port-number;
   routing-instance instance;
   tag-list tag-list;
   target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
   notify-filter profile-name;
   parameters {
```

```
        message-processing-model (v1 | v2c | v3);
        security-level (authentication | none | privacy);
        security-model (usm | v1 | v2c);
        security-name security-name;
    }
}
```

Related Documentation

- Configuring the SNMPv3 Trap Notification on page 93
- Configuring the Trap Notification Filter on page 94
- Configuring the Trap Target Address on page 95
- Defining and Configuring the Trap Target Parameters on page 98
- Configuring SNMP Informs on page 101
- Configuring the Remote Engine and Remote User on page 102
- Configuring the Inform Notification Type and Target Address on page 104
- Complete SNMPv3 Configuration Statements on page 76
- Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 9

Summary of SNMP Configuration Statements

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

access-list

Syntax

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

Hierarchy Level [edit snmp routing-instance-access]

Release Information Statement introduced in Junos OS Release 8.4.

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

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

Related Documentation

- routing-instance-access on page 159

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
- Configuring the Agent Address for SNMP Traps on page 59

authorization

Syntax authorization *authorization*;

Hierarchy Level [edit snmp community *community-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 Set the access authorization for SNMP **Get**, **GetBulk**, **GetNext**, and **Set** requests.

Options ***authorization***—Access authorization level:

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

Default: read-only

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

Related Documentation
- Configuring the SNMP Community String on page 54

 Copyright © 2012, Juniper Networks, Inc.

categories

Syntax

```
categories {
   category;
}
```

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

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

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

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

Options *category*—Name of a trap type: **authentication**, **chassis**, **configuration**, **link**, **remote-operations**, **rmon-alarm**, **routing**, **sonet-alarms**, **startup**, or **vrrp-events**.

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

Related Documentation
- Configuring SNMP Trap Groups on page 60

client-list

Syntax

```
client-list client-list-name {
   ip-addresses;
}
```

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 ***client-list-name***—Name of the client list.

ip-addresses—IP addresses of the SNMP clients to be added to the client list,

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

Related Documentation
- Adding a Group of Clients to an SNMP Community on page 55

client-list-name

Syntax client-list-name *client-list-name*;

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

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 FX Series switches.

Description Add a client list or prefix list to an SNMP community.

Options ***client-list-name***—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
- Adding a Group of Clients to an SNMP Community on page 55

clients

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

Hierarchy Level [edit snmp community *community-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 FX 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
- Configuring the SNMP Community String on page 54

 Copyright © 2012, Juniper Networks, Inc.

commit-delay

Syntax commit-delay *seconds*;

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

- Configuring the Commit Delay Timer on page 52

community

Syntax

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

Hierarchy Level

[edit snmp]

Release Information

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

Description

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

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

Default

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

Options

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

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring the SNMP Community String on page 54

Copyright © 2012, Juniper Networks, Inc.

contact

Syntax contact *contact*;

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
- Configuring the System Contact on a Device Running Junos OS on page 50

description

Syntax description *description*;

Hierarchy Level [edit snmp]

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

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

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

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

Related Documentation
- Configuring the System Description on a Device Running Junos OS on page 51

Copyright © 2012, Juniper Networks, Inc.

destination-port

Syntax destination-port *port-number*;

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 ***port-number***—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
- Configuring SNMP Trap Groups on page 60

engine-id

See engine-id

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
- Filtering Duplicate SNMP Requests on page 51

 Copyright © 2012, Juniper Networks, Inc.

filter-interfaces

Syntax

```
filter-interfaces {
  interfaces {
    all-internal-interfaces;
    interface 1;
    interface 2;
  }
}
```

Hierarchy Level [edit snmp]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 9.4 for EX Series Switches.

Description Filter out information related to specific interfaces from the output of SNMP **Get** and **GetNext** requests performed on interface-related MIBs.

Options

all-internal-interfaces—Filters out information from SNMP **Get** and **GetNext** requests for the specified interfaces.

interfaces—Specifies the interfaces to filter out from the output of SNMP **Get** and **GetNext** requests.

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

Related Documentation

- Filtering Interface Information Out of SNMP Get and GetNext Output on page 65

interface

Syntax interface [*interface-names*];

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 ***interface-names***—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
- Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 63

location

Syntax location *location*;

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 ***location***—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
- Configuring the System Location for a Device Running Junos OS on page 50

 Copyright © 2012, Juniper Networks, Inc.

logical-system

Syntax

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

Hierarchy Level

[edit snmp community *community-name*],
[edit snmp trap-group],
[edit snmp trap-options]
[edit snmp v3target-address *target-address-name*]

Release Information

Statement introduced in Junos OS Release 9.3
Statement introduced in Junos OS Release 9.0 for EX Series switches.

NOTE: **The logical-system statement replaces the logical-router statement, and is backward-compatible with Junos OS Release 8.3 and later.**

Description

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

Include at the **[edit snmp trap-options]** hierarchy level to specify a logical-system address as the source address of an SNMP trap.

Include at the **[edit snmp v3 target-address]** hierarchy level to specify a logical-system name as the destination address for an SNMPv3 trap or inform.

Options

logical-system-name—Name of the logical system.

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

Required Privilege Level

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

Related Documentation

- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184
- Configuring the Trap Target Address on page 95

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
- Trap Support for Routing Instances

name

Syntax name *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 Set the system name from the command-line interface.

Options *name*—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
- Configuring the System Name on page 52

 Copyright © 2012, Juniper Networks, Inc.

nonvolatile

Syntax

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

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

- Configuring the Commit Delay Timer on page 52
- commit-delay on page 149

oid

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

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

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

- Configuring MIB Views on page 66

routing-instance

Syntax

routing-instance *routing-instance-name*;

Hierarchy Level

[edit snmp community *community-name*],
[edit snmp community *community-name* logical-system *logical-system-name*],
[edit snmp trap-group *group*]

Release Information

Statement introduced in Junos OS Release 8.3.
Added to the **[edit snmp community *community-name*]** hierarchy level in Junos OS Release 8.4.
Added to the **[edit snmp community *community-name* logical-system *logical-system-name*]** hierarchy level in Junos OS Release 9.1.
Statement introduced in Junos OS Release 9.1 for EX Series switches.

Description

Specify a routing instance for SNMPv1 and SNMPv2 trap targets. All targets configured in the trap group use this routing instance.

If the routing instance is defined within a logical system, include the **logical-system *logical-system-name*** statement at the **[edit snmp community *community-name*]** hierarchy level and specify the **routing-instance** statement under the **[edit snmp community *community-name* logical-system *logical system-name*]** hierarchy level.

Options

routing-instance-name—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

- Configuring SNMP Trap Groups on page 60
- Configuring the Source Address for SNMP Traps on page 58
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184

Copyright © 2012, Juniper Networks, Inc.

routing-instance-access

Syntax

```
[edit snmp]
  routing-instance-access {
    access-list {
      routing-instance;
      routing-instance 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
- Enabling SNMP Access over Routing Instances on page 183

snmp

Syntax snmp { ... }

Hierarchy Level [edit]

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

Description Configure SNMP.

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

Related Documentation
- Configuring SNMP on a Device Running Junos OS on page 48

source-address

Syntax source-address *address*;

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

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

- Configuring the Source Address for SNMP Traps on page 58

 Copyright © 2012, Juniper Networks, Inc.

targets

Syntax

```
targets {
   address;
}
```

Hierarchy Level

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

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

address—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

- Configuring SNMP Trap Groups on page 60

traceoptions

Syntax

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

Hierarchy Level [edit snmp]

Release Information

Statement introduced before Junos OS Release 7.4.
file *filename* option added in Junos OS Release 8.1.
world-readable | no-world-readable option added in Junos OS Release 8.1.
match *regular-expression* option added in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description

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

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

Options

file *filename*—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.

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

Range: 2 through 1000 files

Default: 10 files

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

- **all**—Log all SNMP events.
- **configuration**—Log reading of configuration at the **[edit snmp]** hierarchy level.

Copyright © 2012, Juniper Networks, Inc.

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

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

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

Range: 10 KB through 1 GB

Default: 1000 KB

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

Required Privilege Level

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

Related Documentation

- Tracing SNMP Activity on a Device Running Junos OS on page 67

trap-group

Syntax

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

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

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

The remaining statements are explained separately.

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

Related Documentation

- Configuring SNMP Trap Groups on page 60

 Copyright © 2012, Juniper Networks, Inc.

trap-options

Syntax

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

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

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.

The remaining statements are explained separately.

Default

Disabled

Required Privilege Level

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

Related Documentation

- Configuring SNMP Trap Options on page 57

version

Syntax version (all | v1 | v2);

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

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

- Configuring SNMP Trap Groups on page 60

 Copyright © 2012, Juniper Networks, Inc.

view

See the following sections:

- view (Associating a MIB View with a Community) on page 167
- view (Configuring a MIB View) on page 168

view (Associating a MIB View with a Community)

Syntax view *view-name*;

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

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

Description Associate a view with a community. A view represents a group of MIB objects.

Options *view-name*—Name of the view. You must use a view name already configured in the **view** statement at the **[edit snmp]** hierarchy level.

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

Related Documentation

- Configuring the SNMP Community String on page 54

view (Configuring a MIB View)

Syntax

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

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 *community-name*]** hierarchy level.

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

Options

view-name—Name of the view.

The remaining 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

- Configuring MIB Views on page 66
- Associating MIB Views with an SNMP User Group on page 87
- **community on page 150**

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 10

SNMP Remote Operations

- SNMP Remote Operations Overview on page 169
- Using the Ping MIB for Remote Monitoring Devices Running Junos OS on page 172
- Starting a Ping Test on page 172
- Monitoring a Running Ping Test on page 174
- Gathering Ping Test Results on page 177
- Stopping a Ping Test on page 178
- Interpreting Ping Variables on page 178

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" on page 491 and "Traceroute MIB" on page 549 topics in the *Junos OS SNMP MIBs and Traps Reference*.

This topic covers the following sections:

- SNMP Remote Operation Requirements on page 170
- Setting SNMP Views on page 170
- Setting Trap Notification for Remote Operations on page 170
- Using Variable-Length String Indexes on page 171
- Enabling Logging on page 172

Copyright © 2012, Juniper Networks, Inc.

SNMP Remote Operation Requirements

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

Setting SNMP Views

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

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

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

Example: Setting SNMP Views

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

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

For more information about the **community** statement, see "Configuring the SNMP Community String" on page 54 and **community**.

For more information about the **view** statement, see "Configuring MIB Views" on page 66, **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.

Copyright © 2012, Juniper Networks, Inc.

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 60.

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 67.

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

Related Documentation

- Using the Ping MIB for Remote Monitoring Devices Running Junos OS on page 172
- Using the Traceroute MIB for Remote Monitoring Devices Running Junos OS

Using the Ping MIB for Remote Monitoring Devices Running Junos OS

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

RFC 2925 is the authoritative description of the Ping MIB in detail and provides the ASN.1 MIB definition of the Piing MIB.

Related Documentation

- SNMP Remote Operations Overview on page 169
- Starting a Ping Test on page 172
- Monitoring a Running Ping Test on page 174
- Gathering Ping Test Results on page 177
- Stopping a Ping Test on page 178
- Interpreting Ping Variables on page 178

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**

Copyright © 2012, Juniper Networks, Inc.

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 170.

There are two ways to start a ping test:

- Using Multiple Set Protocol Data Units (PDUs) on page 173
- Using a Single Set PDU on page 173

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**

Copyright © 2012, Juniper Networks, Inc.

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 174
- pingProbeHistoryTable on page 175
- Generating Traps on page 176

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**.

Copyright © 2012, Juniper Networks, Inc.

- 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 175.

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 60 and "Example: Setting Trap Notification for Remote Operations" on page 171.

Copyright © 2012, Juniper Networks, Inc.

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 174. For more information about Ping MIB traps, see "Generating Traps" on page 176.

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 11 on page 177.

Table 11: 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 12 on page 178.

Copyright © 2012, Juniper Networks, Inc.

Table 12: 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 13 on page 178.

Table 13: 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:

Copyright © 2012, Juniper Networks, Inc.

- **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.

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 11

SNMP Support for Routing Instances

- Understanding SNMP Support for Routing Instances on page 181
- Identifying a Routing Instance on page 182
- Enabling SNMP Access over Routing Instances on page 183
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184
- Example: Configuring Interface Settings for a Routing Instance on page 185
- Configuring Access Lists for SNMP Access over Routing Instances on page 186
- Trap Support for Routing Instances on page 187
- MIB Support Details on page 187

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 182). Service providers can use this information for their own management needs or export the data for use by their customers.

Copyright © 2012, Juniper Networks, Inc.

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
- Trap Support for Routing Instances
- Identifying a Routing Instance on page 182
- Enabling SNMP Access over Routing Instances on page 183
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184
- Configuring Access Lists for SNMP Access over Routing Instances on page 186

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

Copyright © 2012, Juniper Networks, Inc.

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 181
- Enabling SNMP Access over Routing Instances on page 183
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184

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 181
- Identifying a Routing Instance on page 182
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184
- Configuring Access Lists for SNMP Access over Routing Instances on page 186

Copyright © 2012, Juniper Networks, Inc.

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 181
- Identifying a Routing Instance on page 182
- Enabling SNMP Access over Routing Instances on page 183
- Configuring Access Lists for SNMP Access over Routing Instances on page 186
- Example: Configuring Interface Settings for a Routing Instance on page 185

Copyright © 2012, Juniper Networks, Inc.

Example: Configuring Interface Settings for a Routing Instance

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

```
[edit chassis]
aggregated-devices {
   ethernet {
      device-count 5;
   }
}
[edit interfaces ae0]
vlan-tagging;
aggregated-ether-options {
   minimum-links 2;
   link-speed 100m;
}
unit 0 {
   vlan-id 100;
   family inet {
      address 10.1.0.1/24;
   }
}
[edit interfaces fe-1/1/0]
fastether-options {
   802.3ad ae0;
}
[edit interfaces fe-1/1/1]
fastether-options {
   802.3ad ae0;
}
[edit routing-instances]
INFrtd {
   instance-type virtual-router;
   interface fe-1/1/0.0;
   interface fe-1/1/1.0;
   interface fe-1/1/5.0;
   interface ae0.0;
   protocols {
      ospf {
         area 0.0.0.0 {
            interface all;
         }
      }
   }
}
```

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

```
router# snmpwalk -Os router1 INFrtd@public dot3adAggTable
dot3adAggMACAddress.59 = 0:90:69:92:93:f0
dot3adAggMACAddress.65 = 0:90:69:92:93:f0
dot3adAggActorSystemPriority.59 = 0
```

```
dot3adAggActorSystemPriority.65 = 0
dot3adAggActorSystemID.59 = 0:0:0:0:0:0
dot3adAggActorSystemID.65 = 0:0:0:0:0:0
dot3adAggAggregateOrIndividual.59 = true(1)
dot3adAggAggregateOrIndividual.65 = true(1)
dot3adAggActorAdminKey.59 = 0
dot3adAggActorAdminKey.65 = 0
dot3adAggActorOperKey.59 = 0
dot3adAggActorOperKey.65 = 0
dot3adAggPartnerSystemID.59 = 0:0:0:0:0:0
dot3adAggPartnerSystemID.65 = 0:0:0:0:0:0
dot3adAggPartnerSystemPriority.59 = 0
dot3adAggPartnerSystemPriority.65 = 0
dot3adAggPartnerOperKey.59 = 0
dot3adAggPartnerOperKey.65 = 0
dot3adAggCollectorMaxDelay.59 = 0
dot3adAggCollectorMaxDelay.65 = 0
```

Related Documentation

- Understanding SNMP Support for Routing Instances on page 181
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184

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.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Understanding SNMP Support for Routing Instances on page 181
- Enabling SNMP Access over Routing Instances on page 183
- Specifying a Routing Instance in an SNMPv1 or SNMPv2c Community on page 184

Trap Support for Routing Instances

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

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

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

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

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

Related Documentation

- Understanding SNMP Support for Routing Instances on page 181
- MIB Support Details on page 187

MIB Support Details

Table 14 on page 187 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 14: MIB Support for Routing Instances (Juniper Networks MIBs)

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

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

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

Copyright © 2012, Juniper Networks, Inc.

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

Object	Support Class	Description/Notes
jnxCos(15) jnxCosIfqStatsTable(1) jnxCosFcTable(2) jnxCosFcIdTable(3) jnxCosQstatTable(4)	Class 3	Objects 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.

Copyright © 2012, Juniper Networks, Inc.

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

Object	Support Class	Description/Notes
jnxVpnMIB(26)	Class 2	All instances within a logical system 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 15 on page 191 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.

Copyright © 2012, Juniper Networks, Inc.

Table 15: 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 15: 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**

Copyright © 2012, Juniper Networks, Inc.

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

Class	MIB	Objects
Class 1	**rfc2863a.mib**	Examples: **ifXtable** **ifStackTable**
	rfc2665a.mib	**etherMIB**
	rfc2515a.mib	**atmMIB** objects Examples: **atmInterfaceConfTable** **atmVplTable** **atmVclTable**
	rfc2465.mib	**ip-v6mib** Examples: **ipv6IfTable** **ipv6AddrPrefixTable** **ipv6NetToMediaTable** **ipv6RouteTable**
	rfc2787a.mib	**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 15: Class 1 MIB Objects (Standard and Juniper MIBs) *(continued)*

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

Table 16 on page 195 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.

Copyright © 2012, Juniper Networks, Inc.

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

Class	MIB	Objects
Class 2	**rfc3813.mib**	**mplsLsrStdMIB** Examples: **mplsInterfaceTable** **mplsInSegmentTable** **mplsOutSegmentTable** **mplsLabelStackTable** **mplsXCTable** (and related MIB objects)
	igmpmib.mib	**igmpStdMIB**
	l3vpnmib.mib	**mplsVpnmib**
	jnx-mpls.mib	Example: **mplsLspList**
	jnx-ldp.mib	**jnxLdp** Example: **jnxLdpStatsTable**
	jnx-vpn.mib	**jnxVpnMIB**
	jnx-bgpmib2.mib	**jnxBgpM2Experiment**

Table 17 on page 196 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 17: Class 3 MIB Objects (Standard and Juniper MIBs)

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

Table 18 on page 197 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.

Copyright © 2012, Juniper Networks, Inc.

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

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

Related Documentation

- Understanding SNMP Support for Routing Instances on page 181
- Trap Support for Routing Instances on page 187

Copyright © 2012, Juniper Networks, Inc.

PART 3

SRX1400, SRX3400 and SRX3600 Services Gateways MIBs

- SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs on page 201
- Interpreting the Structure of Management Information MIB on page 207
- Interpreting the Enterprise-Specific AAA Objects MIB on page 213
- Interpreting the Enterprise-Specific Access Authentication Objects MIB on page 219
- Interpreting the Enterprise-Specific Alarm MIB on page 223
- Interpreting the Enterprise-Specific ATM Class-of-Service MIB on page 227
- Interpreting the Enterprise-Specific BGP4 V2 MIB on page 235
- Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB on page 237
- Interpreting the Enterprise-Specific Chassis MIBs on page 239
- Interpreting the Enterprise-Specific Chassis Cluster MIBs on page 365
- Interpreting the Enterprise-Specific Configuration Management MIB on page 369
- Interpreting the Enterprise-Specific Destination Class Usage MIB on page 375
- Interpreting the Enterprise-Specific DNS Objects MIB on page 379
- Interpreting the Enterprise-Specific Ethernet MAC MIB on page 381
- Interpreting the Enterprise-Specific Event MIB on page 383
- Interpreting the Enterprise-Specific Firewall MIB on page 385
- Interpreting the Enterprise-Specific Host Resources MIB on page 389
- Interpreting the Enterprise-Specific Interface MIB on page 391
- Interpreting the Enterprise-Specific IP Forward MIB on page 397
- Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object MIB on page 399
- Interpreting the Enterprise-Specific IPsec Monitoring MIB on page 415
- Interpreting the Enterprise-Specific IPv4 MIB on page 423
- Interpreting the Enterprise-Specific IPv6 MIB on page 425
- Interpreting the Enterprise-Specific License MIB on page 433
- Interpreting the Enterprise-Specific Logical Systems (LSYS) MIBs on page 437

- Interpreting the Enterprise-Specific Network Address Translation Objects MIB on page 479
- Interpreting the Enterprise-Specific Packet Forwarding Engine MIB on page 487
- Interpreting the Enterprise-Specific Ping MIB on page 491
- Interpreting the Enterprise-Specific Policy Objects MIB on page 507
- Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB on page 513
- Interpreting the Enterprise-Specific RMON Events and Alarms MIB on page 515
- Interpreting the Enterprise-Specific Security Interface Extension Objects MIB on page 519
- Interpreting the Enterprise-Specific Security Screening Objects MIB on page 523
- Interpreting the Enterprise-Specific SNMP IDP MIB on page 537
- Interpreting the Enterprise-Specific Source Class Usage MIB on page 539
- Interpreting the Enterprise-Specific SPU Monitoring MIB on page 541
- Interpreting the Enterprise-Specific System Log MIB on page 545
- Interpreting the Enterprise-Specific Traceroute MIB on page 549
- Interpreting the Enterprise-Specific Utility MIB on page 551
- Interpreting the VPN Certificate Objects MIB on page 555

 Copyright © 2012, Juniper Networks, Inc.

CHAPTER 12

SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs

This chapter contains the following topic:

- List of SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs on page 201

List of SRX1400, SRX3400, and SRX3600 Services Gateways Supported Enterprise-Specific MIBs

Junos OS supports the following enterprise-specific MIBs:

- Structure of Management Information MIB—Contains object identifiers (OIDs) for the security branch of the MIBs used in Junos OS for J Series and SRX Series devices product, services and traps. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. It also 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/junos11.2/topics/reference/mibs/mib-jnx-js-smi.txt. For more information, see "Structure of Management Information MIB" on page 207.
- AAA Objects MIB—Provides support for monitoring user authentication, authorization, and accounting through the RADIUS, LDAP, SecurID, and local authentication servers. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-user-aaa.txt .

  For more information, see "AAA Objects MIB" on page 213.
- Access Authentication Objects MIB—Provides support for monitoring firewall authentication, including data about the users trying to access firewall-protected resources and the firewall authentication service itself. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-auth.txt .

Copyright © 2012, Juniper Networks, Inc.

For more information, see "Access Authentication Objects MIB" on page 219.

- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-chassis-alarm.txt .

  For more information, see "Alarm MIB" on page 223.

- ATM CoS MIB—Provides support for monitoring Asynchronous Transfer Mode, version 2 (ATM2) virtual circuit (VC) class-of-service (CoS) configurations. It also provides CoS queue statistics for all VCs that have CoS configured. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-atm-cos.txt .

  For more information, see "ATM Class-of-Service MIB" on page 227.

- 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/junos11.2/topics/reference/mibs/mib-jnx-bgpmib2.txt .

  For more information, see "BGP4 V2 MIB" on page 235.

- BFD 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/junos11.2/topics/reference/mibs/mib-jnx-bfd.txt .

  For more information, see "Bidirectional Forwarding Detection MIB" on page 237.

- Chassis MIB—Provides support for environmental monitoring (power supply state, board voltages, fans, temperatures, and air flow) and inventory support for the chassis, System Control Board (SCB), System and Switching Board (SSB), Switching and Forwarding Model (SFM), Flexible PIC Concentrators (FPCs), and PICs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/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/junos11.4/topics/reference/mibs/mib-jnx-jsrpd.txt .

  For more information, see "Chassis Cluster MIB" on page 365.

- 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/junos11.2/topics/reference/mibs/mib-jnx-cfgmgmt.txt .

 Copyright © 2012, Juniper Networks, Inc.

For more information, see "Configuration Management MIB" on page 369.

- Destination Class Usage MIB—Provides support for monitoring packet counts based on the ingress and egress points for traffic transiting your networks. Ingress points are identified by input interface. Egress points are identified by destination prefixes grouped into one or more sets, known as destination classes. One counter is managed per interface per destination class, up to a maximum of 16 counters per interface. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-dcu.txt .

  For more information, see "Destination Class Usage MIB" on page 375.

- DNS Objects MIB—Provides support for monitoring DNS proxy queries, requests, responses, and failures. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-dns.txt .

  For more information, see "DNS Objects MIB" on page 379.

- 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/junos11.2/topics/reference/mibs/mib-jnx-mac.txt .

  For more information, see "Ethernet MAC MIB" on page 381.

- 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/junos11.2/topics/reference/mibs/mib-jnx-event.txt .

  For more information, see "Event MIB" on page 383.

- 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/junos11.2/topics/reference/mibs/mib-jnx-firewall.txt .

  For more information, see "Firewall MIB" on page 385.

- Host Resources MIB—Extends the **hrStorageTable** object, providing a measure of the usage of each file system on the router in percentage. Previously, the objects in the **hrStorageTable** measured the usage in allocation units—**hrStorageUsed** and **hrStorageAllocationUnits**—only. Using the percentage measurement, you can more easily monitor and apply thresholds on usage. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-hostresources.txt .

  For more information, see "Host Resources MIB" on page 389.

- 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/junos11.2/topics/reference/mibs/mib-jnx-if-extensions.txt .

  For more information, see "Interface MIB" on page 391.

- IP Forward MIB—Extends the standard IP Forwarding Table MIB (RFC 2096) to include CIDR forwarding information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-ipforward.txt .

  For more information, see "IP Forward MIB" on page 397.

- 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/junos11.2/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt .

  For more information, see "IPSec Monitoring MIB" on page 415.

- IPsec Generic Flow Monitoring Object MIB—Based on **jnx-ipsec-monitor-mib**, this MIB provides support for monitoring IPsec and IPsec VPN management objects. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt .

  For more information, see "IPsec Generic Flow Monitoring Object MIB" on page 399.

- 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/junos11.2/topics/reference/mibs/mib-jnx-ipv4.txt .

  For more information, see "IPv4 MIB" on page 423.

- 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/junos11.2/topics/reference/mibs/mib-jnx-license.txt .

  For more information, see "License MIB" on page 433.

- Logical Systems MIB—Provides support for logical systems security profile. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.4/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt.

  For more information, see "Logical Systems MIB" on page 437.

- Network Address Translation (NAT) Objects MIB—Provides support for monitoring network address translation (NAT). This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-nat.txt .

 Copyright © 2012, Juniper Networks, Inc.

For more information, see "NAT Objects MIB" on page 479.

- 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/junos11.2/topics/reference/mibs/mib-jnx-pfe.txt .

  For more information, see "Packet Forwarding Engine MIB" on page 487.

- 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/junos11.2/topics/reference/mibs/mib-jnx-ping.txt .

  For more information, see "PING MIB" on page 491.

- Policy Objects MIB—Provides support for monitoring the security policies that control the flow of traffic from one zone to another. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-policy.txt .

  For more information, see "Policy Objects MIB" on page 507.

- 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/junos11.2/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" on page 513.

- 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/junos11.2/topics/reference/mibs/mib-jnx-rmon.txt .

  For more information, see "RMON Events and Alarms MIB" on page 515.

- Security Interface Extension Objects MIB—Provides support for the security management of interfaces. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-if-ext.txt .

  For more information, see "Security Interface Extension Objects MIB" on page 519.

- Security Screening Objects MIB—Defines the MIB for the Juniper Networks Enterprise Firewall screen functionality. This MIB is currently supported only by Junos OS for J

Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-screening.txt .

For more information, see "Security Screening Objects MIB" on page 523.

- 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/junos11.2/topics/reference/mibs/mib-jnx-scu.txt .

  For more information, see "Source Class Usage MIB" on page 539.

- 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/junos11.2/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt .

  For more information, see "SPU Monitoring Objects MIB" on page 541.

- 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/junos11.2/topics/reference/mibs/mib-jnx-syslog.txt .

  For more information, see "System Log MIB" on page 545.

- 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/junos11.2/topics/reference/mibs/mib-jnx-traceroute.txt .

  For more information, see "Traceroute MIB" on page 549.

- Utility MIB—Provides SNMP support for exposing Junos OS data and has tables that contain information on each type of data, such as integer and string. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-util.txt .

  For more information, see "Utility MIB" on page 551.

- VPN Certificate Objects MIB—Provides support for monitoring the local and CA certificates loaded on the router. This MIB is currently supported only by Junos OS for J Series and SRX Series devices. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-js-cert.txt .

  For more information, see "VPN Certificate Objects MIB" on page 555.

Related Documentation

- Structure of Management Information MIB on page 207

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 13

Interpreting the Structure of Management Information MIB

- Structure of Management Information MIB on page 207
- jnxProducts on page 208
- jnxServices on page 208
- jnxMibs on page 209
- jnxTraps on page 211
- jnxExperiment on page 212

Structure of Management Information MIB

The Juniper Networks enterprise-specific Structure of Management Information MIB defines the top-level structure of the Juniper Networks enterprise-specific MIB space. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-smi.txt .

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

- jnxProducts on page 208
- jnxServices on page 208
- jnxMibs on page 209
- jnxTraps on page 211
- jnxExperiment on page 212

Related Documentation

- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxProducts

The object identifier for the **jnxProducts** root branch of the Structure of Management Information MIB is **{juniperMIB 1}**. This branch of the MIB describes the Juniper Networks routers and their components, such as product line, product name, model, number of slots, and media space for holding PICs. It also provides information about the system's power supply state, board voltages, fans, temperatures, and air flow. In general, this branch of the Structure of Management Information MIB is rarely polled for information because it is descriptive. However, you can poll this branch of the Structure of Management Information MIB to determine the **sysObjectId** of a router as defined by MIB-II.

jnxProducts 1 is reserved for Junos OS based products.

Related Documentation

- Structure of Management Information MIB on page 207
- jnxServices on page 208
- jnxMibs on page 209
- jnxTraps on page 211
- jnxExperiment on page 212

jnxServices

The object identifier for the **jnxServices** root branch is **{juniperMIB 2}**. This MIB file added the nodes to create the Juniper Networks security tree structure under the object node **jnxJsObjects**. In general, the prefix **jnxJs** is used to name the object identifiers and to designate them. This branch of the network describes the Juniper Networks services objects that provide enhanced network security. This MIB is currently supported by Junos OS for J Series and SRX Series devices only.

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

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

Copyright © 2012, Juniper Networks, Inc.

The Juniper Networks enterprise-specific security MIBs include:

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

For more information about these MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 31.

Related Documentation

- Structure of Management Information MIB on page 207
- jnxProducts on page 208
- jnxMibs on page 209
- jnxTraps on page 211
- jnxExperiment on page 212

jnxMibs

The object identifier for the **jnxMibs** root branch is **{juniperMIB 3}** and includes one main subbranch, **jnxBoxAnatomy**, whose object identifier is **{jnxMibs 1}**. The other Juniper Networks enterprise-specific MIBs are also branches of **jnxMibs**. These Juniper Networks enterprise-specific MIBs include:

- MPLS MIB—Whose object identifier is **{jnxMibs 2}**.
- Juniper Networks enterprise-specific extensions to the Interface MIB—Whose object identifier is **{jnxMibs 3}**.
- Alarm MIB—Whose object identifier is **{jnxMibs 4}**.
- Firewall MIB—Whose object identifier is **{jnxMibs 5}**.
- Destination Class Usage MIB—Whose object identifier is **{jnxMibs 6}**.
- Juniper Networks enterprise-specific extensions to the Ping MIB—Whose object identifier is **{jnxMibs 7}**.
- Juniper Networks enterprise-specific extensions to the Traceroute MIB—Whose object identifier is **{jnxMibs 8}**.

Copyright © 2012, Juniper Networks, Inc.

- ATM MIB—Whose object identifier is **{jnxMibs 10}**.
- IPv6 and ICMPv6 MIB—Whose object identifier is **{jnxMibs 11}**.
- IPv4 MIB—Whose object identifier is **{jnxMibs 12}**.
- Juniper Networks enterprise-specific extensions to the RMON Events and Alarms MIB—Whose object identifier is **{jnxMIBs 13}**.
- Juniper Networks enterprise-specific extensions to the LDP Traps MIB—Whose object identifier is **{jnxMibs 14}**.
- Class-of-Service MIB—Whose object identifier is **{jnxMibs 15}**.
- Source Class Usage MIB—Whose object identifier is **{jnxMibs 16}**.
- Reverse-Path-Forwarding MIB—Whose object identifier is **{jnxMibs 17}**.
- Configuration Management MIB—Whose object identifier is **{jnxMibs 18}**.
- Passive Monitoring MIB—Whose object identifier is **{jnxMibs 19}**.
- SONET/SDH Interface Management MIB—Whose object identifier is **{jnxMibs 20}**.
- ATM Class-of-Service MIB—Whose object identifier is **{jnxMibs 21}**.
- IPsec Monitoring MIB—Whose object identifier is **{jnxMibs 22}**.
- Ethernet MAC MIB—Whose object identifier is **{jnxMibs 23}**.
- SONET APS MIB—Whose object identifier is **{jnxMibs 24}**.
- Chassis Definitions for Router Model MIB—Whose object identifier is **{jnxMibs 25}**.
- VPN MIB—Whose object identifier is **{jnxMibs 26}**.
- Flow Collection Services MIB—Whose object identifier is **{jnxMibs 28}** .
- RSVP Traffic Engineering (TE) MIB—Whose object identifier is **{jnxMibs 30}**.
- Host Resources MIB—Whose object identifier is **{jnxMibs 31}**.
- Services PIC MIB—Whose object identifier is **{jnxMibs 32}**.
- Dynamic Flow Capture (DFC) MIB—Whose object identifier is **{jnxMibs 33}**.
- Chassis Forwarding MIB—Whose object identifier is **{jnxMibs 34}**.
- System Log MIB—Whose object identifier is **{jnxMibs 35}**.
- MPLS LDP MIB—Whose object identifier is **{jnxMibs 36}**.
- Event MIB—Whose object identifier is **{jnxMibs 37}**.
- IP Forward MIB—Whose object identifier is **{jnxMibs 38}**.
- **jnxJSMibRoot**—Whose object identifier is **{jnxMibs 39}**.
- **jnxEXMibRoot**—Whose object identifier is **{jnxMibs 40}**.
- **jnxWXMibRoot**—Whose object identifier is **{jnxMibs 41}**.
- Packet Forwarding Engine MIB—Whose object identifier is **{jnxMibs 44}**.
- BFD MIB—Whose object identifier is **{jnxMibs 45}**.

 Copyright © 2012, Juniper Networks, Inc.

- Utility MIB—Whose object identifier is **{jnxMibs 47}**.
- L2ALD MIB—Whose object identifier is **{jnxMibs 48}**.
- L2TP MIB—Whose object identifier is **{jnxMibs 49}**.
- RPM MIB—Whose object identifier is **{jnxMibs 50}**.
- User AAA MIB—Whose object identifier is **{jnxMibs 51}**.
- Generic IPsec MIB—Whose object identifier is **{jnxMibs 52}**.
- L2CP MIB—Whose object identifier is **{jnxMibs 53}**.
- Pseudowire TDM MIB—Whose object identifier is **{jnxMibs 54}**.
- Pseudowire TC MIB—Whose object identifier is **{jnxMibs 55}**.
- OTN MIB—Whose object identifier is **{jnxMibs 56}**.

For more information about these MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 31.

Related Documentation

- Structure of Management Information MIB on page 207
- jnxProducts on page 208
- jnxServices on page 208
- jnxTraps on page 211
- jnxExperiment on page 212

jnxTraps

The object identifier for the **jnxTraps** root branch of the Structure of Management Information MIB is **{juniperMIB 4}**. The **jnxTraps** root branch contains the enterprise-specific SNMP traps supported by the Junos OS. These Juniper Networks enterprise-specific SNMP traps include:

- **jnxChassisTraps**—Whose object identifier is **{jnxTraps 1}**.
- **jnxChassisOKTraps**—Whose object identifier is **{jnxTraps 2}**.
- **jnxRmonTraps**—Whose object identifier is **{jnxTraps 3}**.
- **jnxLdpTraps**—Whose object identifier is **{jnxTraps 4}**.
- **jnxCmNotifications**—Whose object identifier is **{jnxTraps 5}**.
- **jnxSonetNotifications**—Whose object identifier is **{jnxTraps 6}**.
- **jnxPMonNotifications**— Whose object identifier is **{jnxTraps 7}**
- **jnxCollectorNotifications**—Whose object identifier is **{jnxTraps 8}**.
- **jnxPingNotification**—Whose object identifier is **{jnxTraps 9}**.
- **jnxSpNotification**—Whose object identifier is **{jnxTraps10}**.

Copyright © 2012, Juniper Networks, Inc.

- **jnxDfcNotifications**—Whose object identifier is **{jnxTraps11}**.
- **jnxSyslogNotifications**—Whose object identifier is **{jnxTraps12}**.
- **jnxEventNotifications**—Whose object identifier is **{jnxTraps13}**.
- **jnxVccpNotifications**—Whose object identifier is **{jnxTraps14}**.
- **jnxOtnNotifications**—Whose object identifier is **{jnxTraps15}**.

Related Documentation

- Structure of Management Information MIB on page 207
- jnxProducts on page 208
- jnxServices on page 208
- jnxMibs on page 209
- jnxExperiment on page 212
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

jnxExperiment

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

jnxExperiment MIBs are defined as the following MIBs:

- IETF work-in-process MIBs that have not been assigned a permanent object identifier by the IANA.
- Juniper Networks work-in-process MIBs that have not achieved final production quality or field experience.

The following draft supports the **jnxExperiment** MIB space: Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version* (**jnxBgpM2PrefixInPrefixes**, **jnxBgpM2PrefixInPrefixesAccepted**, and **jnxBgpM2PrefixInPrefixesRejected** objects only).

Related Documentation

- Structure of Management Information MIB on page 207
- jnxProducts on page 208
- jnxServices on page 208
- jnxMibs on page 209
- jnxTraps on page 211

 Copyright © 2012, Juniper Networks, Inc.

CHAPTER 14

Interpreting the Enterprise-Specific AAA Objects MIB

- AAA Objects MIB on page 213
- AAA Objects MIB Textual Conventions on page 214
- jnxUserAAAStatTable on page 216
- jnxUserAAAServerName on page 216
- Access Authentication-Related Traps on page 217

AAA Objects MIB

The Juniper Networks enterprise-specific AAA Objects MIB, whose object ID is **{jnxUserAAAMibRoot 1}**, defines the objects pertaining to user authentication, authorization, and accounting.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-user-aaa.txt .

Object IDs for next branch nodes are as follows:

- jnxUserAAAGlobalStats—**{jnxUserAAAObjects 1}**
- jnxUserAAAAccessAuthStats—**{jnxUserAAAObjects 2}**
- jnxUserAAATrapVars—**{jnxUserAAAObjects 3}**
- jnxUserAAAAssignment—**{jnxUserAAAObjects 4}**
- jnxUserAAAAccessProfile—**{jnxUserAAAObjects 5}**

Object IDs for statistic counters related to access authentication are as follows:

- jnxTotalAuthenticationRequests—**{jnxUserAAAGlobalStats 1}**
- jnxTotalAuthenticationResponses—**{jjnxUserAAAGlobalStats 2}**

The object ID for address pools is as follows:

Copyright © 2012, Juniper Networks, Inc.

- **jnxUserAAAAccessPool 1—{jnxUserAAAAccessPoolTable 1}**

For information about the enterprise-specific AAA Objects MIB, see the following topics

- AAA Objects MIB Textual Conventions on page 214
- jnxUserAAAStatTable on page 216
- jnxUserAAAServerName on page 216
- Access Authentication-Related Traps on page 217
- jnxUserAAAAccessPool

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

AAA Objects MIB Textual Conventions

jnxAuthenticateType—Defines the method to authenticate a user.

Table 19: jnxAuthenticateType

Method	Syntax Integer
none—no authentication	(0)
radius—authentication through a RADIUS server	(1)
local—local authenticaiton	(2)
ldap—authentication through an LDAP server	(3)
securid—authentication through an RSA SecurID authentication server	(4)
jsrc—authentication through JSRC	(5)

jnxaccountingType—Defines the accounting method to authenticate a user.

Table 20: jnxAccountingType

Method	Syntax Integer
none—no accounting	(0)
radius—accounting through a RADIUS server	(1)
local—local accounting	(2)
ldap—accounting through an LDAP server	(3)

Copyright © 2012, Juniper Networks, Inc.

Table 20: jnxAccountingType *(continued)*

Method	Syntax Integer
securid—accounting through an RSA SecurID authentication server	(4)
jsrc—accounting through JSRC	(5)

jnxAuthorizationType—Defines the method to authorize a user.

Table 21: jnxAuthorizationType

Method	Syntax Integer
none—no authentication	(0)
radius—authorization through a RADIUS server	(1)
local—local authorization	(2)
ldap—authorization through an LDAP server	(3)
securid—authorization through an RSA SecurID authentication server	(4)
jsrc—authorization through JSRC	(5)

jnxProvisioningType—Defines the method to provision a user.

Table 22: jnxProvisioningType

Method	Syntax Integer
none—no provisioning	(0)
radius—provisioning through a RADIUS server	(1)
local—local provisioning	(2)
ldap—provisioning through an LDAP server	(3)
securid—provisioning through an RSA SecurID authentication server	(4)
jsrc—provisioning through JSRC	(5)

Related Documentation

- AAA Objects MIB on page 213
- jnxUserAAAStatTable on page 216
- jnxUserAAAServerName on page 216

- jnxUserAAAAccessPool
- Access Authentication-Related Traps on page 217

jnxUserAAAStatTable

jnxUserAAAStatTable, whose object ID is **{jnxUserAAAAccessAuthStats 1}**, exposes the user authentication statistics listed in Table 23 on page 216.

Table 23: jnxUserAAAStatTable

Object	Object ID	Description
jnxUserAAAStatEntry	**jnxUserAAAStatTable 1**	Statistics entries collected for authentication. Sequence of parameters: • **jnxUserAAAStatAuthType** • **jnxUserAAAStatRequestReceived** • **jnxUserAAAStatAccessAccepted** • **jnxUserAAAStatAccessRejected**
jnxUserAAAStatAuthType	**jnxUserAAAStatEntry 1**	Indicates the authentication type. This entry uniquely identifies the statistics counters related to its authentication.
jnxUserAAAStatRequestReceived	**jnxUserAAAStatEntry 2**	The number of the request received.
jnxUserAAAStatAccessAccepted	**jnxUserAAAStatEntry 3**	The number of the access granted. This entry is an aggregated statistic for this type of authentication.
jnxUserAAAStatAccessRejected	**jnxUserAAAStatEntry 4**	This number of the access request rejected. This entry is an aggregated statistic for this type of authentication.

Related Documentation

- AAA Objects MIB on page 213
- AAA Objects MIB Textual Conventions on page 214
- jnxUserAAAServerName on page 216
- jnxUserAAAAccessPool
- Access Authentication-Related Traps on page 217

jnxUserAAAServerName

jnxUserAAAServerName, whose object ID is **{jnxUserAAAAccessAuthStats 1}**, specifies the server name that identifies the authentication server.

Related Documentation

- AAA Objects MIB on page 213
- AAA Objects MIB Textual Conventions on page 214
- jnxUserAAAStatTable on page 216

Copyright © 2012, Juniper Networks, Inc.

- jnxUserAAAAccessPool
- Access Authentication-Related Traps on page 217

Access Authentication-Related Traps

Table 24 on page 217 identifies access authentication traps.

Table 24: Access Authentication-Related Traps

Object	Object ID	Description
jnxAccessAuthServiceUp	**jnxUserAAANotifications 1**	Access authentication trap to signify that the specified service has started.
jnxAccessAuthServiceDown	**jnxUserAAANotifications 2**	Access authentication trap to signify that the specified service has been stopped.
jnxAccessAuthServerDisabled	**jnxUserAAANotifications 3**	Access authentication trap to signify that the external authentication server is not responding.
jnxAccessAuthServerEnabled	**jnxUserAAANotifications 4**	Access authentication trap to signify that the external authentication server started responding again.

Related Documentation

- AAA Objects MIB on page 213
- AAA Objects MIB Textual Conventions on page 214
- jnxUserAAAStatTable on page 216
- jnxUserAAAServerName on page 216
- jnxUserAAAAccessPool
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 15

Interpreting the Enterprise-Specific Access Authentication Objects MIB

- Access Authentication Objects MIB on page 219
- jnxJsFwAuthStats on page 219
- jnxJsAuthTrapVars on page 220
- jnxJsAuthNotifications on page 221

Access Authentication Objects MIB

The Juniper Networks enterprise-specific Access Authentication Objects MIB, whose object ID is **{jnxJsAuth 1}**, defines the objects that pertain to access authentication. Firewall and security features restrict the accessing of protected resources (ideally on different zones) behind a firewall based on their source IP and other credentials.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-auth.txt .

For information about the enterprise-specific Access Authentication Objects MIB, see the following topics:

- jnxJsFwAuthStats on page 219
- jnxJsAuthTrapVars on page 220
- jnxJsAuthNotifications on page 221

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsFwAuthStats

jnxJsFwAuthStats, whose object ID is **{jnxJsAuthObjects 1}**, exposes the firewall authentication statistics listed in Table 25 on page 220.

Table 25: jnxJsFwAuthStats

Object	Object ID	Description
jnxJsFwAuthNumPendingUsers	jnxJsFwAuthStats 1	Number of users currently waiting to be authenticated by the firewall user authentication mechanism.
jnxJsFwAuthNumSuccUsers	jnxJsFwAuthStats 2	Number of users currently allowed access by the firewall user authentication mechanism.
jnxJsFwAuthNumFailedUsers	jnxJsFwAuthStats 3	Number of users who currently failed to be authenticated by the firewall user authentication mechanism.
jnxJsFwAuthTotalUsers	jnxJsFwAuthStats 4	Total number of users that are accessing or attempting to access resources managed by the firewall user authentication mechanism.

Related Documentation

- Access Authentication Objects MIB on page 219
- jnxJsAuthTrapVars on page 220
- jnxJsAuthNotifications on page 221

jnxJsAuthTrapVars

jnxJsAuthTrapVars identifies access authentication traps variables listed in Table 26 on page 220.

Table 26: jnxJsAuthTrapVars

Object	Object ID	Description
jnxJsFwAuthUserName	jnxJsAuthTrapVars 1	Name of the user who is attempting to be authenticated or has been authenticated.
jnxJsFwAuthServiceDesc	jnxJsAuthTrapVars 2	Service or application name that the authentication is performed for: Telnet, FTP, HTTP.
jnxJsFwAuthReason	jnxJsAuthTrapVars 3	Reason for the trap being generated—authentication failure due to: timeout, invalid password, invalid username, and so on.
jnxJsFwAuthClientIpAddr	jnxJsAuthTrapVars 4	Authentication client's IP address.

Related Documentation

- Access Authentication Objects MIB on page 219
- jnxJsFwAuthStats on page 219
- jnxJsAuthNotifications on page 221

Copyright © 2012, Juniper Networks, Inc.

jnxJsAuthNotifications

jnxJsAuthNotifications, whose object ID is **{jnxJsAuthMIB 0}**, identifies the user access authentication notifications listed in Table 27 on page 221 .

Table 27: jnxJsAuthNotifications

Object	Object ID	Description
jnxJsFwAuthFailure	**jnxJsAuthNotifications** 1	A firewall user authentication status trap to signify whether a user using the pass-through firewall authentication mechanism has been rejected due to the reason specified in the trap. • **jnxJsFwAuthUserName** is the user. • **jnxClientIPAddress** is the IP address the user came from. • **jnxJsFwAuthServiceDesc** specifies the application by which the authentication was performed. • **jnxJsFwAuthReason** indicates the reason for failure.
jnxJsFwAuthServiceUp	**jnxJsAuthNotifications** 2	Firewall user authentication service has started.
jnxJsFwAuthServiceDown	**jnxJsAuthNotifications** 3	Firewall user authentication service has stopped.
jnxJsFwAuthCapacityExceeded	**jnxJsAuthNotifications** 4	Firewall user authentication maximum capacity has been exceeded. **jnxJsFwAuthTotalUsers** indicates the total number of users being authenticated, and it has exceeded the maximum allowable users.

Related Documentation

- Access Authentication Objects MIB on page 219
- jnxJsFwAuthStats on page 219
- jnxJsAuthTrapVars on page 220

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 16

Interpreting the Enterprise-Specific Alarm MIB

- Alarm MIB on page 223
- jnxAlarmRelayMode on page 223
- jnxYellowAlarms on page 224
- jnxRedAlarms on page 225

Alarm MIB

The Juniper Networks enterprise-specific Alarm MIB, whose object identifier is **{jnxMibs 4}**, contains information about alarms from the router chassis.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-chassis-alarm.txt .

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

- jnxAlarmRelayMode on page 223
- jnxYellowAlarms on page 224
- jnxRedAlarms on page 225

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxAlarmRelayMode

The **jnxAlarmRelayMode**, whose object identifier is **{jnxCraftAlarms 1}**, denotes the alarm relay mode of the craft interface panel for the yellow and red alarms. This object uses the following integer values:

- **1–Other**: Other or unknown state
- **2–passOn**: Alarms are passed on. The alarm relay is activated to pass on the yellow or red alarms to audible sirens or visual flashing devices.

- **3–cutOff**: Alarms are turned off. Both the yellow and red alarms are cut off from the alarm relays and do not get passed on to audible sirens or visual flashing devices.

NOTE: Both the yellow and red alarms can be cut off from the alarm relay using a Alarm Cutoff/Lamp Test button on the front panel of the router chassis.

Related Documentation

- Alarm MIB on page 223
- jnxYellowAlarms on page 224
- jnxRedAlarms on page 225

jnxYellowAlarms

The **jnxYellowAlarms**, whose object identifier is **{jnxCraftAlarms 2}**, contains the objects listed in Table 28 on page 224.

Table 28: jnxYellowAlarms

Object	Object ID	Description
jnxYellowAlarmState	**jnxYellowAlarms 1**	Denotes the yellow alarm state on the craft interface panel of the router chassis or on the front of the switch chassis. This object contains one of the following integer values: • **other–1**: The alarm state is unknown. • **off–2**: The yellow alarms are off. • **on–3**: The yellow alarms are on.
jnxYellowAlarmCount	**jnxYellowAlarms 2**	Shows the number of currently active and nonsilent yellow alarms. NOTE: EX8200 Virtual Chassis support these alarms only on the XRE200 External Routing Engines and not on the member switches. Standalone EX8200 switches support these alarms.
jnxYellowAlarmLastChange	**jnxYellowAlarms 3**	Shows the value of the **sysUp** time when the state of the yellow alarm last changed from on to off or vice versa. This object returns **0** if the alarm state has not changed after the **sysUp** time was reset last time, or if the value is unknown.

Related Documentation

- Alarm MIB on page 223
- jnxAlarmRelayMode on page 223
- jnxRedAlarms on page 225

Copyright © 2012, Juniper Networks, Inc.

jnxRedAlarms

The **jnxRedAlarms**, whose object identifier is **{jnxCraftAlarms 3}**, contains the objects listed in Table 29 on page 225.

Table 29: jnxRedAlarms

Object	Object ID	Description
jnxRedAlarmState	**jnxRedAlarms 1**	Denotes the state of red alarms on the craft interface panel of the router chassis or on the front of the switch chassis. This object contains one of the following values: • **1–other**: The red alarm state is unknown. • **2–off**: The red alarm is turned off. • **3–on**: The red alarm is on. Typically, the red alarm is on when there is a system failure, power failure, or hardware malfunction; or when a threshold value is exceeded.
jnxRedAlarmCount	**jnxRedAlarms 2**	Shows the number of currently active and nonsilent red alarms. NOTE: EX8200 Virtual Chassis support these alarms only on the XRE200 External Routing Engines and not on the member switches. Standalone EX8200 switches support these alarms.
jnxRedAlarmLastChange	**jnxRedAlarms 3**	Shows the value of the **sysUp** time when the red alarm last changed from on to off or vice versa. This object contains **0** value, if the alarm state has not changed after the **sysUp** time was reset last time, or if the value is unknown.

Related Documentation

- Alarm MIB on page 223
- jnxAlarmRelayMode on page 223
- jnxYellowAlarms on page 224

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 17

Interpreting the Enterprise-Specific ATM Class-of-Service MIB

- ATM Class-of-Service MIB on page 227
- jnxCosAtmVcTable on page 228
- jnxCosAtmVcScTable on page 228
- jnxCosAtmVcQstatsTable on page 230
- jnxCosAtmTrunkTable on page 231

ATM Class-of-Service MIB

The Juniper Networks enterprise-specific ATM Class-of-Service (CoS) MIB provides information about the ATM CoS infrastructure.

The Juniper Networks enterprise-specific ATM CoS MIB uses the following objects and definitions as per the RFCs and MIBs:

- **ifIndex** (RFC 2233, *IF MIB*)
- **atmVclvpi** and **atmVclVci** (RFC 2515, *ATM MIB*)
- **jnxMibs** (*Juniper Networks enterprise-specific SMI MIB*) and **jnxCoSFcIf** (*Juniper Networks enterprise-specific CoS MIB*)

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-atm-cos.txt .

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

- jnxCosAtmVcTable on page 228
- jnxCosAtmVcScTable on page 228
- jnxCosAtmVcQstatsTable on page 230
- jnxCosAtmTrunkTable on page 231

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxCosAtmVcTable

The **jnxCosAtmVcTable**, whose object identifier is **{jnxAtmCos 1}**, contains information about virtual circuits (VC) that have CoS configured.

Each **jnxCosAtmVcEntry**, whose object identifier is **{jnxCosAtmVcTable 1}**, is indexed using **ifIndex**, **atmVclVpi**, and **atmVclVci**, and contains the **jnxCosAtmVcCosMode** object. The **jnxCosAtmVcCosMode** object represents the mode of CoS queue priority for the VC.

There are two modes, **strict** mode and **alternate** mode:

- **Strict** mode (represented by integer value **0**): One of the four queues has high priority, and is always serviced before the other three queues. The remaining queues are serviced in a round robin fashion.
- **Alternate** mode (represented by integer value **1**): Even though one of the four queues has high priority, the servicing of the queues alternates between the high priority queue and the other queues.

Related Documentation

- ATM Class-of-Service MIB on page 227
- jnxCosAtmVcScTable on page 228
- jnxCosAtmVcQstatsTable on page 230
- jnxCosAtmTrunkTable on page 231

jnxCosAtmVcScTable

The **jnxCosAtmVcScTable**, whose object identifier is **{jnxAtmCos 2}**, contains ATM-scheduler configuration parameters for each forwarding class within a specified VC.

NOTE: **The schedulers mentioned in this MIB are specific to an ATM interface, and are different from the typical schedulers specified using the CoS configuration command-line interface (CLI).**

Each **jnxCosAtmVcScEntry**, whose object identifier is **{jnxCosAtmVcScTable 1}**, represents ATM-scheduler configuration parameters per forwarding class and per VC, and is indexed using **ifIndex**, **atmVclVpi**, and **atmVclVci**. Table 30 on page 229 lists the ATM scheduler parameters as represented by a **jnxCosAtmVcScEntry**.

 Copyright © 2012, Juniper Networks, Inc.

Table 30: jnxCosAtmVcScTable

Object	Object ID	Description
jnxCosAtmVcScPriority	**jnxCosAtmVcScEntry 1**	Represents the ATM scheduler priority for the queue associated with the specified forwarding class within the VC.
jnxCosAtmVcScTxWeightType	**jnxCosAtmVcScEntry 2**	Represents the ATM scheduler transmit-weight-type for the queue associated with the specified forwarding class inside the VC. The transmit-weight-type is represented either as the number of cells or as a percentage of the queue size.
jnxCosAtmVcScTxWeight	**jnxCosAtmVcScEntry 3**	Represents the transmit weight of the ATM scheduler for the queue associated with the specified forwarding class and the VC. **jnxCosAtmVcScTxWeight** is expressed either as the number of cells or as a percentage of the total VC bandwidth. The value of **jnxCosAtmVcScTxWeightType**.determines the unit used.
jnxCosAtmVcScDpType	**jnxCosAtmVcScEntry 4**	Shows the type of RED drop profile configured for the specified forwarding class within the VC. A scheduler can specify either linear or constant drop profile. A constant type drop profile (also known as EPD) specifies that all the cells should be dropped when the number of queued cells exceeds a threshold. A linear type drop profile specifies that only a percentage of cells be dropped based on the number of queued cells at any time.
jnxCosAtmVcScLrdpQueueDepth	**jnxCosAtmVcScEntry 5**	Represents the maximum queue size in cells, as specified by the linear RED drop profile associated with the specified forwarding class within the VC. This object is valid only when the value of the object **jnxCosAtmVcScDpType** is **linearRed(0).**
jnxCosAtmVcScLrdpLowPlpThresh	**jnxCosAtmVcScEntry 6**	Represents the threshold percentage of fill-level beyond which the low packet loss priority (PLP) packets belonging to the specified forwarding class within the VC are randomly dropped. This value is specified by the linear RED drop profile configuration. This object is valid only when the object **jnxCosAtmVcScDpType** is set to **linearRed(0)**.

Copyright © 2012, Juniper Networks, Inc.

Table 30: jnxCosAtmVcScTable *(continued)*

Object	Object ID	Description
jnxCosAtmVcScLrdpHighPlpThresh	**jnxCosAtmVcScEntry 7**	Represents the threshold percentage of the fill level beyond which high PLP packets belonging to the specified forwarding class within the VC are randomly dropped. This value is specified by the linear RED drop profile configuration. This object is valid only when the object **jnxCosAtmVcScDpType** is set to **linearRed(0)**.
jnxCosAtmVcEpdThreshold	**jnxCosAtmVcScEntry 8**	Shows the EPD drop threshold configured for the specified forwarding class within the VC. When the number of cells queued exceeds the value specified for this object, all the cells in the queue are dropped. This object is valid only when the **jnxCosAtmVcScDpType** object is set to **epd(1)**.

Related Documentation

- ATM Class-of-Service MIB on page 227
- jnxCosAtmVcTable on page 228
- jnxCosAtmVcQstatsTable on page 230
- jnxCosAtmTrunkTable on page 231

jnxCosAtmVcQstatsTable

The **jnxCosAtmVcQstatsTable**, whose object identifier is **{jnxAtmCos 3}**, contains queue statistics for VCs and forwarding classes. Each **jnxCosAtmVcQstatsEntry** in the **jnxCosAtmVcQstatsTable** contains the queue status information for a particular forwarding class and VC. The **jnxCosAtmVcQstatsEntry** object uses **ifIndex**, **atmVclVpi**, **atmVclVci**, and **jnxCosFcId** for indexing.

Each **jnxCosAtmVcQstatsEntry** contains the objects listed in Table 31 on page 230.

Table 31: jnxCosAtmVcQstatsTable

Object	Object ID	Description
jnxCosAtmVcQstatsOutPackets	**jnxCosAtmVcQstatsEntry 1**	Represents the number of packets belonging to a particular forwarding class that is transmitted on a specific VC.
jnxCosAtmVcQstatsOutBytes	**jnxCosAtmVcQstatsEntry 2**	Represents the number of bytes of a particular forwarding class that are transmitted on a specific VC.
jnxCosAtmVcQstatsOutRedDropPkts	**jnxCosAtmVcQstatsEntry 3**	Represents the number of RED-dropped outgoing packets of a particular forwarding class that are transmitted on a specific VC.

Copyright © 2012, Juniper Networks, Inc.

Table 31: jnxCosAtmVcQstatsTable *(continued)*

Object	Object ID	Description
jnxCosAtmVcQstatsOutNonRedDrops	**jnxCosAtmVcQstatsEntry 4**	Represents the number of outgoing packets, of a particular forwarding class and transmitted on a specific VC, that are dropped because of errors in packets.
jnxCosAtmVcQstatsOutLpBytes	**jnxCosAtmVcQstatsEntry 5**	Represents the number of low PLP (PLP0) bytes transmitted.
jnxCosAtmVcQstatsOutLpPkts	**jnxCosAtmVcQstatsEntry 6**	Represents the number of low PLP(PLP0) packets that are transmitted.
jnxCosAtmVcQstatsOutLpDropBytes	**jnxCosAtmVcQstatsEntry 7**	Represents the number of low PLP (PLP0) bytes dropped at the output queue.
jnxCosAtmVcQstatsOutHpDropBytes	**jnxCosAtmVcQstatsEntry 8**	Represents the number of high PLP (PLP1) bytes dropped at the output queue.
jnxCosAtmVcQstatsOutLpDropPkts	**jnxCosAtmVcQstatsEntry 9**	Represents the number of low PLP (PLP0) packets dropped at the output queue.
jnxCosAtmVcQstatsOutHpDropPkts	**jnxCosAtmVcQstatsEntry 10**	Represents the number of high PLP (PLP1) packets dropped at the output queue.

Related Documentation

- ATM Class-of-Service MIB on page 227
- jnxCosAtmVcTable on page 228
- jnxCosAtmVcScTable on page 228
- jnxCosAtmTrunkTable on page 231

jnxCosAtmTrunkTable

The **jnxCosAtmTrunkTable**, whose object identifier is **{jnxAtmCos 4}**, contains statistics and configuration information related to the ATM trunk CoS interface.

The **jnxCosAtmTrunkEntry**, whose object identifier is **{jnxCosAtmTrunkTable 1}**, uses **ifIndex** and **jnxCosFcId**, and contains the objects listed in Table 32 on page 232.

Table 32: jnxCosAtmTrunkTable

Object	Object ID	Description
jnxCosAtmTrunkMode	**jnxCosAtmTrunkEntry** 1	Represents the mode of CoS queue priority for the trunk: • **Strict** mode (represented by integer value **0**): One of the four queues has high priority, and is always serviced before the other three queues. The remaining queues are serviced in a round-robin fashion. • **Alternate** mode (represented by integer value **1**): Even though one of the four queues has high priority, the servicing of the queues alternates between the high priority queue and the other queues.
jnxCosAtmTrunkScPriority	**jnxCosAtmTrunkEntry** 2	Represents the ATM scheduler priority for the queue associated with a particular forwarding class within the trunk.
jnxCosAtmTrunkScTxWeightType	**jnxCosAtmTrunkEntry** 3	Represents the ATM scheduler transmit weight type for the queue associated with a particular forwarding class inside the trunk. The weight type can be expressed either as the number of cells or as a percentage of the queue size.
jnxCosAtmTrunkScTxWeight	**jnxCosAtmTrunkEntry** 4	Represents the transmit weight for the queue. The transmit weight can be expressed either as the number of cells or as a percentage of the total trunk bandwidth. The unit is determined by the value set for **jnxCosAtmTrunkScTxWeightType**.
jnxCosAtmTrunkQaType	**jnxCosAtmTrunkEntry** 5	Represents the ATM queue admission type used for the specified trunk. Available values for this object are: **red**(**1**), **singleEpd**(**2**), and **dualEpd**(**3**).
jnxCosAtmTrunkEpdThresholdPlp0	**jnxCosAtmTrunkEntry** 6	Represents the threshold value beyond which all PLP0 cells get dropped. This object has a valid value only when the value for **jnxCosAtmTrunkQaType** is set to **singleEpd** or **dualEpd**.
jnxCosAtmTrunkEpdThresholdPlp1	**jnxCosAtmTrunkEntry** 7	Represents the threshold value beyond which all PLP1 cells get dropped. This object has a valid value only when the **jnxCosAtmTrunkQaType** object is set to **dualEpd**.
jnxCosAtmTrunkQstatsOutPackets	**jnxCosAtmTrunkEntry** 8	Represents the number of packets that belong to a particular forwarding class, and are transmitted on the specific trunk.
jnxCosAtmTrunkQstatsOutBytes	**jnxCosAtmTrunkEntry** 9	Represents the number of bytes that belong to a particular forwarding class, and are transmitted on the specific trunk.
jnxCosAtmTrunkQstatsOutDrops	**jnxCosAtmTrunkEntry** 10	Represents the number of outgoing packets on the trunk that are dropped.
jnxCosAtmTrunkQstatsOutLpBytes	**jnxCosAtmTrunkEntry** 11	Represents the number of low PLP (PLP0) bytes that are transmitted on the trunk.

Copyright © 2012, Juniper Networks, Inc.

Table 32: jnxCosAtmTrunkTable *(continued)*

Object	Object ID	Description
jnxCosAtmTrunkQstatsOutLpPkt	jnxCosAtmTrunkEntry 12	Represents the number of low PLP (PLP0) packets that are transmitted on the trunk.
jnxCosAtmTrunkQstatsOutLpDropBytes	jnxCosAtmTrunkEntry 13	Represents the number of low PLP (PLP0) bytes dropped at the output queue.
jnxCosAtmTrunkQstatsOutHpDropBytes	jnxCosAtmTrunkEntry 14	Represents the number of high PLP (PLP1) bytes that are dropped at the output queue.
jnxCosAtmTrunkQstatsOutLpDropPkts	jnxCosAtmTrunkEntry 15	Represents the number of low PLP (PLP0) packets that are dropped at the output queue.
jnxCosAtmTrunkQstatsOutHpDropPkts	jnxCosAtmTrunkEntry 16	Represents the number of high PLP (PLP1) packets dropped at the output queue.
jnxCosAtmTrunkQstatsOutHpBytes	jnxCosAtmTrunkEntry 17	Represents the number of high PLP (PLP1) bytes that are transmitted on the trunk.
jnxCosAtmTrunkQstatsOutHpPkts	jnxCosAtmTrunkEntry 18	Represents the number of high PLP (PLP1) packets that are transmitted on the trunk.

Related Documentation

- ATM Class-of-Service MIB on page 227
- jnxCosAtmVcTable on page 228
- jnxCosAtmVcScTable on page 228
- jnxCosAtmVcQstatsTable on page 230

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 18

Interpreting the Enterprise-Specific BGP4 V2 MIB

- BGP4 V2 MIB on page 235
- jnxBgpM2PrefixCountersTable on page 236

BGP4 V2 MIB

The Juniper Networks enterprise-specific BGP version 4 (BGP4) V2 MIB, whose object identifier is **{jnxBgpM2Experiment 1}**, contains objects used to monitor BGP peer-received prefix counters. It is based upon similar objects in the MIB documented in Internet draft draft-ietf-idr-bgp4-mibv2-03.txt, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4), Second Version*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-bgpmib2.txt .

NOTE:

For the BGP4 V2 MIB, the Junos OS supports only the following objects:

- **jnxBgpM2PrefixInPrefixes**
- **jnxBgpM2PrefixInPrefixesAccepted**
- **jnxBgpM2PrefixInPrefixesRejected**

For information about the enterprise-specific BGP4 V2 MIB objects, see the following topics:

- jnxBgpM2PrefixCountersTable on page 236
- Juniper Networks Enterprise-Specific BGP Traps on page 140

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxBgpM2PrefixCountersTable

jnxBgpM2PrefixCountersTable contains counters associated with a BGP peer.

jnxBgpM2PrefixCountersEntry contains information about the prefix counters of a BGP peer, and the objects listed in Table 33 on page 236.

Table 33: jnxBgpM2PrefixCountersEntry

Object	Object Identifier	Description
jnxBgpM2PrefixInPrefixes	jnxBgpM2PrefixCountersEntry 7	The total number of prefixes received from a peer.
jnxBgpM2PrefixInPrefixesAccepted	jnxBgpM2PrefixCountersEntry 8	The total number of prefixes received from a peer that are eligible to be active in the routing table.
jnxBgpM2PrefixInPrefixesRejected	jnxBgpM2PrefixCountersEntry 9	The total number of prefixes received from a peer that are not eligible to be active in the routing table.

Related Documentation

- BGP4 V2 MIB on page 235
- Juniper Networks Enterprise-Specific BGP Traps on page 140

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 19

Interpreting the Enterprise-Specific Bidirectional Forwarding Detection (BFD) MIB

- Bidirectional Forwarding Detection MIB on page 237
- jnxBfdSessTable on page 237
- Notifications for the BFD MIB on page 238

Bidirectional Forwarding Detection MIB

The Juniper Networks enterprise-specific Bidirectional Forwarding Detection (BFD) MIB, whose object identifier is **{jnxBfdMibRoot 1}**, sends the current operational status for the transmit interval and detection time of BFD sessions. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-bfd.txt .

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

- jnxBfdSessTable on page 237
- Notifications for the BFD MIB on page 238

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxBfdSessTable

The **jnxBfdSessTable**, whose object identifier is **{jnxBfdObjects 1}**, is an extension to the **jnxSessTable**. It provides information about threshold values for the transmit interval and detection time on BFD sessions.

The **jnxBfdSessEntry**, whose object identifier is **{jnxBfdSessTable 1}**, has four objects, which are listed in Table 34 on page 238.

Table 34: jnxBfdSessTable

Object	Object Identifier	Description
jnxBfdSessThresTxInterval	**jnxBfdSessEntry 1**	The threshold value (in microseconds) for the transmit interval. If the current transmit interval value (**jnxBfdSessCurrTxInterval**) adapts to a value greater than the threshold value, a trap is raised (**jnxBfdSessTxIntervalHigh**).
jnxBfdSessCurrTxInterval	**jnxBfdSessEntry 2**	The current transmit interval for the session (in microseconds).
jnxBfdSessThreshDectTime	**jnxBfdSessEntry 3**	The threshold value (in microseconds) for the detection time. If the current detection time value (**jnxBfdSessCurrDectTime)** adapts to a value greater than the threshold value, a trap is raised (**jnxBfdSessDetectionTimeHigh**).
jnxBfdSessCurrDectTime	**jnxBfdSessEntry 4**	The current detection time for the session (in microseconds).

Related Documentation

- Bidirectional Forwarding Detection MIB on page 237
- Notifications for the BFD MIB on page 238

Notifications for the BFD MIB

Table 35 on page 238 lists the supported notifications for the BFD MIB.

Table 35: Supported Notifications for the BFD MIB

Object	Object Identifier	Description
jnxBfdSessTxIntervalHigh	**jnxBfdNotification 1**	A notification generated when the threshold value for the transmit interval is configured (**jnxBfdSessThresTxInterval**) and the BFD session transmit interval (**jnxBfdSessCurrTxInterval**) adapts to a value greater than the threshold value. This trap is sent only once, when the threshold is first exceeded. The transmit interval can continue to adapt beyond the threshold value.
jnxBfdSessDetectionTimeHigh	**jnxBfdNotification 2**	A notification generated when the threshold value for the detection time is configured (**jnxBfdSessThresDectTime**) and the BFD session detection time (**jnxBfdSessCurrDectTime**) adapts to a value greater than the threshold value. This trap is sent only once, when the threshold is first exceeded. The detection time can continue to adapt beyond the threshold value.

Related Documentation

- Bidirectional Forwarding Detection MIB on page 237
- jnxBfdSessTable on page 237

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 20

Interpreting the Enterprise-Specific Chassis MIBs

- Chassis MIBs on page 239
- Chassis MIB Textual Conventions on page 240
- jnxBoxAnatomy on page 241
- Chassis Traps on page 355
- Chassis Definitions for Router Model MIB on page 360
- MIB Objects for the SRX1400 Services Gateway on page 361
- MIB Objects for the SRX3400 Services Gateway on page 362
- MIB Objects for the SRX3600 Services Gateway on page 363

Chassis MIBs

The enterprise-specific Chassis MIB provides information on the router and its components. MIB objects represent each component and the status of the components. The enterprise-specific Chassis Definitions for Router Model MIB contains the object identifiers (OIDs) that are used by the Chassis MIB to identify platform and chassis components. The Chassis MIB provides information that changes often. The Chassis Definitions for Router Model MIB provides information that changes less often.

You can retrieve information from the MIB using any network management system. For a downloadable version of the Chassis Definitions for Router Model MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-chas-defines.txt .

For more information about the enterprise-specific Chassis MIBs, see the following topics:

- Chassis MIB Textual Conventions on page 240
- jnxBoxAnatomy on page 241
- Chassis Traps on page 355
- Chassis Definitions for Router Model MIB on page 360
- MIB Objects for the SRX1400 Services Gateway on page 361

Copyright © 2012, Juniper Networks, Inc.

- MIB Objects for the SRX3400 Services Gateway on page 362
- MIB Objects for the SRX3600 Services Gateway on page 363

Chassis MIB Textual Conventions

The enterprise-specific Chassis MIB uses the **jnxChassisId** object to denote the router chassis type. **jnxChassisId** can be one of the following integer values:

- **1**—Unknown
- **2**—Single chassis
- **3**—scc (TX Matrix router)
- **4**—lcc0 (T640 router)
- **5**—lcc1 (T640 router)
- **6**—lcc2 (T640 router)
- **7**—lcc3 (T640 router)
- **8**—jcs1
- **9**—jcs2
- **10**—jcs3
- **11**—jcs4
- **12**—sfc0
- **13**—sfc1
- **14**—sfc2
- **15**—sfc3
- **16**—sfc4
- **17**—lcc4
- **18**—lcc5
- **19**—lcc6
- **20**—lcc7
- **21**—lcc8
- **22**—lcc9
- **23**—lcc10
- **24**—lcc11
- **25**—lcc12
- **26**—lcc13

Copyright © 2012, Juniper Networks, Inc.

- **27**—lcc14
- **28**—lcc15

Related Documentation

- Chassis MIBs

jnxBoxAnatomy

The object identifier for the **jnxMIBs** root branch is **{juniperMIB 3}** and includes one main subbranch, **jnxBoxAnatomy**, whose object identifier is **{jnxMibs 1}**.

The **jnxBoxAnatomy** MIB is explained in the following sections:

- Top-Level Objects on page 241
- jnxContainersTable on page 242
- jnxContentsLastChange on page 249
- jnxContentsTable on page 249
- jnxLEDLastChange on page 267
- jnxLEDTable on page 267
- jnxFilledLastChange on page 270
- jnxFilledTable on page 271
- jnxOperatingTable on page 287
- jnxRedundancyTable on page 299
- jnxFruTable on page 306
- jnxBoxKernelMemoryUsedPercent on page 355
- jnxBoxSystemDomainType on page 355

Top-Level Objects

The following branches of the **jnxBoxAnatomy** MIB are top-level objects:

- **jnxBoxClass**—The object identifier for the **jnxBoxClass** object is **{jnxBoxAnatomy 1}**. This object classifies the chassis product line.
- **jnxBoxDescr**—The object identifier for the **jnxBoxDescr** object is **{jnxBoxAnatomy 2}**. This object describes the chassis name and model.
- **jnxBoxSerialNo**—The object identifier for the **jnxBoxSerialNo** object is **{jnxBoxAnatomy 3}**. This object indicates the serial number of the chassis. **jnxBoxSerialNo** remains blank if the serial number is unknown or unavailable.
- **jnxBoxRevision**—The object identifier for the **jnxBoxRevision** object is **{jnxBoxAnatomy 4}**. This object indicates the last revision of the chassis.
- **jnxBoxInstalled**—The object identifier for the **jnxBoxInstalled** object is **{jnxBoxAnatomy 5}**. This object indicates the last time the box was installed and operational, represented by the **sysUpTime** value.

Copyright © 2012, Juniper Networks, Inc.

jnxContainersTable

The object identifier for the **jnxContainersTable** object is **{jnxBoxAnatomy 6}**. This object shows the structure of the chassis.

You can use the **jnxContainersTable** object to retrieve specific information about the router, such as how many of each component the router can contain. For example, the **jnxContainersTable** object of an M20 router indicates that the router can accommodate four Flexible PIC Concentrators (FPCs). However, it does not describe how many FPCs the router actually has.

For more information about how many FPCs are actually on a router, see "jnxContentsTable" on page 249.

Entries within the **jnxContainersTable** object are represented by the **jnxContainersEntry** object, whose object identifier is **{jnxContainersTable 1}**. The **jnxContainersEntry** object contains the following objects that describe the contents of a particular router:

- **jnxContainersIndex**—The index value of an entry in the **jnxContainersEntry** object that corresponds to **jnxContainersType** and **jnxContainersDescr**, and whose object identifier is **{jnxContainersEntry 1}**.
- **jnxContainersView**—The orientation of a container from the front of the router and whose object identifier is **{jnxContainersEntry 2}**. This object also indicates that the container is embedded in the router and how it is accessible from corresponding views. The value of this object is a bitmap represented as a sum. If multiple bits are set, you can access the container from that set of views. The values represent the bit positions and their corresponding views as follows:
  - 1—Front
  - 2—Rear
  - 4—Top
  - 8—Bottom
  - 16—Left side
  - 32—Right side

  For each view plane, if specified counters are scattered in various views, the numbering sequence starts from left to right and then from top to bottom, as follows:
  - Left side
  - Right side
  - Top
  - Bottom

 Copyright © 2012, Juniper Networks, Inc.

- Front
- Rear

NOTE: References to left and right sides are based on the view from the front of the chassis.

In accordance with network management conventions, all indexes in the MIB begin with 1, not 0, although the slot number might be labeled 0.

- **jnxContainersLevel**—The abstraction level of the box or components for the **jnxContainersEntry** object and whose object identifier is **{jnxContainersEntry 3}**. The level is enumerated from the outside to the inside, and from the outer layer to the inner layer.

  For example, if the top level (level 0) of the box refers to the chassis frame, then the next level (level 1) refers to the FPC slot within the chassis frame. Finally, the PIC space within the FPC slot of the chassis corresponds to level 2.

- **jnxContainersWithin**—The container housing the entry at the next-higher level of the **jnxContainersEntry** object and whose object identifier is **{jnxContainersEntry 4}**.

  For example, the within value for **jnxMediaCardSpacePIC.0** is 7. Because the **jnxM20SlotFPC.0** retains an index value of 7, the FPC houses the PIC.

- **jnxContainersType**—The component of the Chassis MIB at a specific index, view, level, and within value for the **jnxContainersEntry** object and whose object identifier is **{jnxContainersEntry 5}**.

- **jnxContainersDescr**—The description of the component in the **jnxContainersEntry** object and whose object identifier is **{jnxContainersEntry 6}**.

- **jnxContainersCount**—The maximum number of a given component that the router can accommodate within the **jnxContainersEntry** object and whose object identifier is **{jnxContainersEntry 7}**.

  For example, the M20 router can house a specific maximum number of FPCs within the chassis frame. The maximum number is not necessarily the actual number of FPCs; this can change dynamically.

Table 36 on page 244 through Table 43 on page 248 provide examples of **jnxContainersEntry** objects in the **jnxContainersTable**. The following column headings for each table are abbreviated to correspond to the parts of the **jnxContainersEntry** objects:

- Index—**jnxContainersIndex**
- View—**jnxContainersView**
- Level—**jnxContainersLevel**
- Within—**jnxContainersWithin**
- Type—**jnxContainersType**

- Description—**jnxContainersDescr**
- Count—**jnxContainersCount**

Table 36 on page 244 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M40 router.

Table 36: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM40.0**	Chassis frame compartment	1
2	2	1	1	**jnxSlotPowerSupply.0**	Power supply compartment	2
3	3	1	1	**jnxSlotCoolingImpeller.0**	Impeller compartment	2
4	2	1	1	**jnxSlotCoolingFan.0**	Fan compartment	3
5	2	1	1	**jnxSlotHostCtrl.0**	Host controller compartment	1
6	1	1	1	**jnxSlotSCB.0**	SCB slot	1
7	1	1	1	**jnxSlotFPC.0**	FPC slot	8
8	1	2	7	**jnxMediaSlotCardPIC.0**	PIC space	4
9	2	1	1	**jnxSlotRoutingEngine.0**	Routing Engine compartment	1

Table 37 on page 244 describes objects in the **jnxContainersTable** of an M20 router.

Table 37: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM20.0**	Chassis frame compartment	1
2	2	1	1	**jnxM20SlotPower.0**	Power supply compartment	2
4	3	1	1	**jnxSlotFan.0**	Fan compartment	4
6	2	1	1	**jnxM20SlotSSB.0**	SSB slot	2
7	1	1	1	**jnxM20SlotFPC.0**	FPC slot	4
8	1	2	7	**jnxM20MediaCardSpacePIC.0**	PIC space	4

Copyright © 2012, Juniper Networks, Inc.

Table 37: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router *(continued)*

Index	View	Level	Within	Type	Description	Count
9	2	1	1	**jnxM20RE.0**	Routing Engine compartment	2
10	1	1	1	**JNXM20FrontPanel.0**	Front display slot	1

Table 38 on page 245 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M160 router.

Table 38: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM160.0**	Chassis frame compartment	1
2	2	1	1	**Jnx160SlotPower.0**	Power supply compartment	2
4	3	1	1	**jnxM160SlotFan.0**	Fan compartment	4
6	2	1	1	**jnxM160SlotSFM.0**	SFM slot	4
7	1	1	1	**jnxM160SlotFPC.0**	FPC slot	8
8	1	2	7	**jnxM160MediaCardSlotPIC.0**	PIC space	4
9	2	1	1	**jnxM160SlotHM.0**	Host slot	2
10	1	1	1	**jnxM160SlotFPM.0**	FPM slot	1
11	2	1	1	**jnxM160SlotPCG.0**	PCG slot	2
12	2	1	1	**jnxM160SlotMCS.0**	MCS slot	2
13	1	1	1	**jnxM160SlotCIP.0**	CIP slot	1

Table 39 on page 245 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M10 router.

Table 39: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM10.0**	Chassis frame compartment	1

Table 39: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router *(continued)*

Index	View	Level	Within	Type	Description	Count
2	2	1	1	**jnxM10SlotPower.0**	Power supply compartment	2
4	2	1	1	**jnxM10SlotFan.0**	Fan compartment	1
6	2	1	1	**jnxM10SlotFEB.0**	FEB slot	1
7	1	1	1	**jnxM10SlotFPC.0**	FPC slot	2
8	1	2	7	**jnxM10MediaCardSpacePIC.0**	PIC space	4
9	2	1	1	**jnxM10SlotRE.0**	Routing Engine compartment	1

Table 40 on page 246 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M5 router.

Table 40: jnxContainersEntry Objects in the jnxContainersTable of an M5 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM5.0**	Chassis frame compartment	1
2	2	1	1	**jnxM5SlotPower.0**	Power supply compartment	2
4	3	1	1	**jnxM5SlotFan.0**	Fan compartment	4
6	2	1	1	**jnxM5SlotFEB.0**	FEB slot	1
7	1	1	1	**jnxM5SlotFPC.0**	FPC slot	1
8	1	2	7	**jnxM5MediaCardSlotPIC.0**	PIC space	4
9	2	1	1	**jnxM5SlotRE.0**	Routing Engine compartment	1

Table 41 on page 247 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of a T640 router.

Copyright © 2012, Juniper Networks, Inc.

Table 41: jnxContainersEntry Objects in the jnxContainersTable of a T640 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisT640.0**	Chassis frame	1
2	2	1	1	**jnxT640SlotPower.0**	PEM slot	2
4	3	1	1	**jnxT640SlotFan.0**	Fan slot	3
7	1	1	1	**jnxT640SlotFPC.0**	FPC slot	8
8	1	2	7	**jnxT640MediaCardSpacePIC.0**	PIC slot	4
9	2	1	1	**jnxT640SlotHM.0**	Host slot	2
10	1	1	1	**jnxT640SlotFPB.0**	FPM slot	1
11	2	1	1	**jnxT640SlotSCG.0**	SCG slot	2
12	2	1	1	**jnxT640SlotCB.0**	CG slot	2
13	1	1	1	**jnxT640SlotCIP.0**	CIP slot	1
14	2	1	1	**jnxT640SlotSPMB.0**	SPMB slot	2
15	2	1	1	**jnxT640SlotSIB.0**	SIB slot	5

Table 42 on page 247 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of a T320 router.

Table 42: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisT320.0**	Chassis frame	1
2	2	1	1	**jnxT320SlotPower.0**	PEM slot	2
4	3	1	1	**jnx320SlotFan.0**	Fan slot	3
7	1	1	1	**jnxT320SlotFPC.0**	FPC slot	8
8	1	2	7	**jnxT320MediaCardSpacePIC.0**	PIC slot	2
9	2	1	1	**jnxT320SlotHM.0**	Host slot	2
10	1	1	1	**jnxT320SlotFPB.0**	FPM slot	1

Copyright © 2012, Juniper Networks, Inc.

Table 42: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router *(continued)*

Index	View	Level	Within	Type	Description	Count
11	2	1	1	**jnxT320SlotSCG.0**	SCG slot	2
12	2	1	1	**jnxT320SlotCB.0**	CB slot	2
13	1	1	1	**jnxT320SlotCIP.0**	CIP slot	1
14	2	1	1	**jnxT320SlotSPMB.0**	SPMB slot	2
15	2	1	1	**jnxT320SlotSIB.0**	SIB slot	3

Table 43 on page 248 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of an M40e router.

Table 43: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisM40e.0**	Chassis frame compartment	1
2	2	1	1	**jnxM40eSlotPower.0**	Power supply compartment	2
4	3	1	1	**jnxM40eSlotFan.0**	Fan compartment	4
6	2	1	1	**jnxM40eSlotSFM.0**	SFM slot	2
7	1	1	1	**jnxM40eSlotFPC.0**	FPC slot	8
8	1	2	7	**jnxM40eMediaCardSpacePIC.0**	PIC space	4
9	2	1	1	**jnxM40eSlotHM.0**	Host slot	2
10	1	1	1	**jnxM40eSlotFPM.0**	FPM slot	1
11	2	1	1	**jnxM40eSlotPCG.0**	PCG slot	2
12	2	1	1	**jnxM40eSlotMCS.0**	MCS slot	2
13	1	1	1	**jnxM40eSlotCIP.0**	CIP slot	1

Table 44 on page 249 describes objects contained in a **jnxContainersEntry** in the **jnxContainersTable** of a T4000 router.

Copyright © 2012, Juniper Networks, Inc.

Table 44: jnxContainersEntry Objects in the jnxContainersTable of a T4000 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	**jnxChassisT4000.0**	Chassis frame	1
2	2	1	1	**jnxT4000SlotPower.0**	PEM slot	2
4	3	1	1	**jnxT4000SlotFan.0**	Fan slot	3
7	1	1	1	**jnxT4000SlotFPC.0**	FPC slot	8
8	1	2	7	**jnxT4000MediaCardSpacePIC.0**	PIC slot	4
9	2	1	1	**jnxT4000SlotHM.0**	RoutingEngine slot	2
10	1	1	1	**jnxT4000SlotFPB.0**	FPM slot	1
11	2	1	1	**jnxT4000SlotSCG.0**	SCG slot	2
12	2	1	1	**jnxT4000SlotCB.0**	CG slot	2
13	1	1	1	**jnxT4000SlotCIP.0**	CIP slot	1
14	2	1	1	**jnxT4000SlotSPMB.0**	SPMB slot	2
15	2	1	1	**jnxT4000SlotSIB.0**	SIB slot	5
19	1	1	1	jnxT4000SlotPSD.0	PSD slot	31

jnxContentsLastChange

The object identifier for **jnxContentsLastChange** object is **{jnxBoxAnatomy 7}**. This object indicates the time at which the box contents last changed, represented by the **sysUpTime** value.

jnxContentsTable

The object identifier for **jnxContentsTable** object is **{jnxBoxAnatomy 8}**. This object specifies the contents of the chassis.

The **jnxContentsTable** lists the contents of an entry, which are defined as follows:

- **jnxContentsContainerIndex**—Associates the **jnxContainersIndex** with the **jnxContainersTable**. The object identifier is **{jnxContentsEntry 1}**.
- **jnxContentsL1Index**—Denotes the level-one index of the container housing the component. The object identifier is **{jnxContentsEntry 2}**. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.

Copyright © 2012, Juniper Networks, Inc.

NOTE: MIBs start with a value of 1, whereas the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- **jnxContentsL2Index**—Denotes the level-two index of the container housing the component. The object identifier is **{jnxContentsEntry 3}**. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- **jnxContentsL3Index**—Denotes the level-three index of the container housing the component. The object identifier is **{jnxContentsEntry 4}**. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- **jnxContentsType**—Indicates the type of the component at a specific container index or L1, L2, or L3 index. The object identifier is **{jnxContentsEntry 5}**.
- **jnxContentsDescr**—Specifies the type of the component in plain English. The object identifier is **{jnxContentsEntry 6}**.
- **jnxContentsSerialNo**—Specfies the serial number of the component. The object identifier is **{jnxContentsEntry 7}**.
- **jnxContentsRevision**—Contains the revision level of the component. The object identifier is **{jnxContentsEntry 8}**.
- **jnxContentsInstalled**—Indicates the time at which the component was last installed and operational, represented by the **sysUpTime** value. The object identifier is **{jnxContentsEntry 9}**.
- **jnxContentsPartNo**—Contains the part number of the component (blank if unknown or unavailable). The object identifier is **{jnxContentsEntry 10}**.
- **jnxContentsChassisId** —Contains the ID of the chassis that contains the components. The object identifier is **{jnxContentsEntry 11}**.
- **jnxContentsChassisDescr**—Specifies the description of the chassis in plain English. The object identifier is **{jnxContentsEntry 12}**.
- **jnxContentsChassisCleiCode**—Contains the CLEI code of the chassis that contains the components listed in the row. A CLEI code is an intelligent code that consists of 10 alphanumeric characters with 4 data elements. The first data element is considered the basic code, with the first two characters indicating the technology or equipment type, and the third and fourth characters denoting the functional subcategory. The second data element represents the features, and its three characters denote functional capabilities or changes. The third data element has one character and denotes a reference to a manufacturer, system ID, specification, or drawing. The fourth data element consists of two characters and contains complementary data. These two characters provide a means of differentiating or providing uniqueness between the eight character CLEI codes by identifying the manufacturing vintage of the product. For more information about CLEI code, see http://www.commonlanguage.com/resources/commonlang/

Copyright © 2012, Juniper Networks, Inc.

productshowroom/showroom/equip_id/carriers/overview.html . The object identifier is **{jnxContentsEntry 13}**.

Table 45 on page 251 through Table 47 on page 258 provide examples of **jnxContentEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxContentsEntry** objects:

- Container index— **jnxContentsContainerIndex**
- L1 Index—**jnxContentsL1Index**
- L2 Index—**jnxContentsL2Index**
- L3 Index—**jnxContentsL3Index**
- Type—**jnxContentsType**
- Description—**jnxContentsDescr**
- Serial Number—**jnxContentsSerialNo**
- Revision—**jnxContentsRevision**
- Installed—**jnxContentsInstalled**
- Part Number—**jnxContentsPartNo**

Table 45 on page 251 provides an example of **jnxContentEntry** objects in the **jnxContentTable** of an M20 router.

Table 45: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	1	0	**jnxBackplaneM20.0**	Midplane	AL3280	REV07	0:0:00:00.00	710-00157
2	1	0	0	**jnxM20PowerDC.0**	DC power supply A	001652	REV 05	0:0:00:00.00	740-00146
2	2	0	0	**jnxM20PowerDC.0**	DC power supply B	001652	REV 05	0:0:00:00.00	740-00146
4	1	0	0	**jnxM20Fan.0**	Front top fan	–	–	0:0:00:00.00	–
4	2	0	0	**jnxM20Fan**	Middle fan	–	–	0:0:00:00.00	–
4	3	0	0	**jnxM20Fan**	Bottom fan	–	–	0:0:00:00.00	–
4	4	4	0	**jnxM20Fan**	Rear fan	–	–	0:0:00:00.00	–
6	1	0	0	**jnxM20SSB.0**	SSB 0 Internet Processor II	AG0809	REV 01	0:0:00:35.17	710-001951

Copyright © 2012, Juniper Networks, Inc.

Table 45: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	1	0	0	**jnxM20FPC.0**	FPC @ 0/*/*	AN1335	REV 01	0:0:01:01.80	710-001292
7	2	0	0	**jnxM20FPC.0**	FPC @ 1/*/*	AN1124	REV 01	0:0:01:07:96	710-001292
7	3	0	0	**jnxM20FPC.0**	FPC @ 2/*/*	AN1726	REV 01	0:0:01:14:12	710-001292
7	4	0	0	**jnxM20FPC.0**	FPC @ 3/*/*	AN1691	REV 01	0:0:01:20.28	710-001292
8	1	1	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 0/0/*	HD4313	REV 04	0:0:00:00.00	750-002992
8	1	2	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 0/1/*	AJ5844	REV 04	0:0:00:00.00	750-002992
8	1	3	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 0/2/*	HD4518	REV 04	0:0:00:00.00	750-002992
8	1	4	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 0/3/*	HD4515	REV 04	0:0:00:00.00	750-002992
8	2	1	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 1/0/*	HD4296	REV 04	0:0:00:00.00	750-002992
8	2	2	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 1/1/*	HD4323	REV 04	0:0:00:00.00	750-002992
8	2	3	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 1/2/*	HD4129	REV 04	0:0:00:00.00	750-002992
8	2	4	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 1/3/*	HD4341	REV 04	0:0:00:00.00	750-002992
8	3	1	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX@ 2/0/*	AH4147	REV 07	0:0:00:00.00	750-002303

Copyright © 2012, Juniper Networks, Inc.

Table 45: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	3	2	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 2/1/*	AH4238	REV 07	0:0:00:00.00	750-002303
8	3	3	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 2/2/*	AH4116	REV 07	0:0:00:00.00	750-002303
8	3	4	0	**jnxM20QuadEther.0**	PIC: 4x, F/E, 100BASE-TX @ 2/3/*	AH4208	REV 07	0:0:00:00.00	750-002303
8	4	1	0	**jnxM20GigEther.0**	PIC: 1x G/E, 1000BASE-SX @ 3/0/*	AS3697	REV 07	0:0:00:00.00	750-001072
8	4	2	0	**jnxM20ChOc12toDS3.0**	PIC: 1x COC12SMIR @ 3/1/*	AE1110	REV 08	0:0:00:00.00	750-001190
8	4	4	0	**jnxM20ChStml1.0**	PIC: 1x CSTM1SMIR @ 3/3/*	AD9599	REV 04	0:0:00:00.00	750-003250
9	1	0	0	**jnxM20RE.0**	Routing Engine	–	–	3:16:16:53.21	–
10	1	0	0	**jnxM20FrontPanel.0**	Front panel display	–	–	0:0:00:00.00	–

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M20 router is as follows:

```
user@host> show chassis hardware
Item    Version  Part number   Serial Number    Description
Chassis53711     M20
Backplane        REV 07   710-001517       AL3280
Power Supply A   REV 05   740-001466          001652      DC
Power Supply B   REV 05   740-001466          001632      DC
Display         REV 04   710-001519      AP9225
Host 0 c900000619e6ba01 teknor
SSB slot 0        REV 01  710-001951        AG0809      Internet Processor
II
FPC 0      REV 01  710-001292  AN1335
 PIC 0     REV 04  750-002992  HD4313      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  AJ5844      4x F/E, 100 BASE-TX
 PIC 2     REV 04  750-002992  HD4518      4x F/E, 100 BASE-TX
 PIC 3     REV 04  750-002992  HD4515      4x F/E, 100 BASE-TX
FPC 1      REV 01  710-001292  AN1124
 PIC 0     REV 04  750-002992  HD4296      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  HD4323      4x F/E, 100 BASE-TX
```

```
  PIC 2      REV 04   750-002992  HD4129      4x F/E, 100 BASE-TX
  PIC 3      REV 04   750-002992  HD4341      4x F/E, 100 BASE-TX
FPC 2        REV 01   710-001292  AN1726
  PIC 0      REV 07   750-002303  AH4147      4x F/E, 100 BASE-TX
  PIC 1      REV 07   750-002303  AH4238      4x F/E, 100 BASE-TX
  PIC 2      REV 07   750-002303  AH4116      4x F/E, 100 BASE-TX
  PIC 3      REV 07   750-002303  AH4208      4x F/E, 100 BASE-TX
FPC 3        REV 01   710-001292  AN1691
  PIC 0      REV 08   750-001072  AS3697      1x G/E, 1000
BASE-SX
  PIC 1      REV 03   750-001190  AE1110      1x COC12, SMIR
  PIC 3      REV 04   750-003250  AD9599      1x CSTM1, SMIR
```

Table 46 on page 254 provides an example of **jnxContentEntry** objects in the **jnxContentTable** of a T640 router.

Table 46: jnxContentsEntry Objects in the jnxContentsTable of a T640 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	**jnxMidplaneT640.0**	Midplane	AX5633	REV 04	0:0:00:00.00	710-002726
2	2	0	0	**jnxT640Power.0**	PEM 1	MD21815	RevX02	0:0:00:00.00	740-002595
4	1	1	0	**jnxT640Fan.0**	Top left front fan	–	–	0:0:00:00.00	–
4	1	2	0	**jnxT640Fan.0**	Top left middle fan	–	–	0:0:00:00.00	–
4	1	3	0	**jnxT640Fan.0**	Top left rear fan	–	–	0:0:00:00.00	–
4	1	4	0	**jnxT640Fan.0**	Top right front fan	–	–	0:0:00:00.00	–
4	1	5	0	**jnxT640Fan.0**	Top right middle fan	–	–	0:0:00:00.00	–
4	1	6	0	**jnxT640Fan.0**	Top right rear fan	–	–	0:0:00:00.00	–
4	2	1	0	**jnxT640Fan.0**	Bottom left front fan	–	–	0:0:00:00.00	–
4	2	2	0	**jnxT640Fan.0**	Bottom left middle fan	–	–	0:0:00:00.00	–
4	2	3	0	**jnxT640Fan.0**	Bottom left rear fan	–	–	0:0:00:00.00	–
4	2	4	0	**jnxT640Fan.0**	Bottom right front fan	–	–	0:0:00:00.00	–

Copyright © 2012, Juniper Networks, Inc.

Table 46: jnxContentsEntry Objects in the jnxContentsTable of a T640 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	2	5	0	**jnxT640Fan.0**	Bottom right middle fan	–	–	0:0:00:00.00	–
4	2	6	0	**jnxT640Fan.0**	Bottom right rear fan	–	–	0:0:00:00.00	–
4	3	1	0	**jnxT640Fan.0**	Fourth blower from top	–	–	0:0:00:00.00	–
4	3	2	0	**jnxT640Fan.0**	Bottom blower	–	–	0:0:00:00.00	–
4	3	3	0	**jnxT640Fan.0**	Middle blower	–	–	0:0:00:00.00	–
4	3	4	0	**jnxT640Fan.0**	Top blower	–	–	0:0:00:00.00	–
4	3	5	0	**jnxT640Fan.0**	Second blower from top	–	–	0:0:00:00.00	–
7	2	0	0	**jnxT640FPC.0**	FPC @ 1/*/*	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	1	0	**jnxT640FPC.0**	FPC @ 1/0/* top temperature sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	2	0	**jnxT640FPC.0**	FPC @ 1/1/* bottom temperature sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	6	0	0	**jnxT640FPC.0**	FPC @ 5/*/*	HD5001	REV 03	0:18:57:02.71	710-001721
7	6	1	0	**jnxT640FPC.0**	FPC @ 5/0/* top temperature sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	6	2	0	**jnxT640FPC.0**	FPC @ 5/1/* bottom temperature sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	8	0	0	**jnxT640FPC.0**	FPC @ 7/*/*	HE3179	REV 01	0:18:56:52.85	710-002385
7	8	1	0	**jnxT640FPC.0**	FPC @ 7/0/* top temperature sensor	HE3179	REV 01	0:18:56:52.85	710-002385

Copyright © 2012, Juniper Networks, Inc.

Table 46: jnxContentsEntry Objects in the jnxContentsTable of a T640 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	8	2	0	**jnxT640FPC.0**	FPC @ 7/1/* bottom temperature sensor	HE3179	REV 01	0:18:56:52.85	710-002385
8	2	1	0	**jnxT640PIC3.0**	PIC: 1x G/E, 1000BASE-SX @ 1/0/*	AP5542	REV 08	0:18:56:50.91	750-001072
8	2	2	0	**jnxT640PIC3.0**	PIC: 1x OC-12 ATM, SMIR @ 1/1/*	AK6894	REV 02	0:18:56:55.24	750-002983
8	2	3	0	**jnxT640PIC3.0**	PIC: 1x G/E, 1000 BASE-SX @ 1/2/*	HD4968	REV 04	0:18:56:55.64	750-001894
8	6	1	0	**jnxT640PIC3.0**	PIC: 1x OC-192 SM SR1 @ 5/0/*	HC0273	REV 01	0:18:57:04.47	750-004535
8	6	2	0	**jnxT640PIC3.0**	PIC: 1x OC-192 SM SR1 @ 5/1/*	HC0271	REV 01	0:18:57:04.55	750-004535
8	6	3	0	**jnxT640PIC3.0**	PIC: 1x OC-192 SM SR1 @ 5/2/*	HC0254	REV 01	0:18:57:04.64	750-004535
8	8	1	0	**jnxT640PIC3.0**	PIC: 2x G/E, 1000Base-SX @ 7/0/*	AD3632	REV 01	0:18:56:55.16	710-002381
8	8	2	0	**jnxT640PIC3.0**	PIC: 4x OC-12 SONET, SMIR @ 7/1/*	AD3831	REV 05	0:18:56:55.18	750-001901
8	8	3	0	**jnxT640PIC3.0**	PIC: 1x OC-48 SONET, SMIR @ 7/2/*	AA9603	REV 01	0:18:56:55.21	750-001900
8	8	4	0	**jnxT640PIC3.0**	PIC: 1x OC-48 SONET, SMSR @ 7/3/*	AD5724	REV 05	0:18:56:55.24	750-001900
9	1	0	0	**jnxT640HM.0**	Host 0	–	–	0:19:19:30.95	–
9	2	0	0	**jnxT640HM.0**	Host 1	2108 6570 0292	REV 01	2:19:45:51.00	740-005022

Copyright © 2012, Juniper Networks, Inc.

Table 46: jnxContentsEntry Objects in the jnxContentsTable of a T640 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
10	1	0	0	**jnxT640FPB.0**	FPM	HE3245	REV 02	0:0:00:00.00	710-002901
11	1	0	0	**jnxT640SCG.0**	SCG 0	HF6023	REV 04	0:0:00:00.00	710-003423
11	2	0	0	**jnxT640SCG.0**	SCG 1	HF6061	REV 04	0:0:00:00.00	710-003423
12	2	0	0	**jnxT640CB.0**	CB 0	HE3614	REV 06	0:0:00:00.00	710-002728
12	2	0	0	**jnxT640CB.0**	CB 1	HE3627	REV 06	0:0:00:00.00	710-002728
13	1	0	0	**jnxT640CIP.0**	CIP	HA4729	REV 05	0:0:00:00.00	710-002895
14	1	0	0	**jnxT640SPMB.0**	SPMB 0	HF6876	REV 02	0:18:56:06.72	710-003229
14	2	0	0	**jnxT640SPMB.0**	SPMB 1	HG6237	REV 02	0:18:56:08.01	710-003229
15	1	0	0	**jnxT640SIB.0**	SIB 0	HJ9669	REV 02	0:0:00:00.00	710-005157
15	2	0	0	**jnxT640SIB.0**	SIB 1	HJ9668	REV 02	0:0:00:00.00	710-005157
15	3	0	0	**jnxT640SIB.0**	SIB 2	HH3039	REV 02	0:0:00:00.00	710-005157
15	4	0	0	**jnxT640SIB.0**	SIB 3	HH3041	REV 02	0:0:00:00.00	710-005157
15	5	0	0	**jnxT640SIB.0**	SIB 4	HJ9657	REV 02	0:0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item       Version Part number Serial number   Description
Chassis    T640
Midplane    REV 04  710-002726  AX5633
FPM GBUS    REV 02  710-002901  HE3245
FPM Display    REV 02  710-002897  HA4873
CIP          REV 05  710-002895  HA4729
PEM 1         RevX02  740-002595  MD21815         Power Entry Module
SCG 0         REV 04  710-003423  HF6023
SCG 1         REV 04  710-003423  HF6061
Host 0    unknown
Host 1    REV 01  740-005022  210865700292   RE-3.0
CB 0          REV 06  710-002728  HE3614
CB 1          REV 06  710-002728  HE3627
FPC 1         REV 01  710-002385  HE3009      FPC Type 1
 CPU          REV 06  710-001726  HC0010
 PIC 0        REV 08  750-001072  AP5542      1x G/E, 1000Base-SX
 PIC 1        REV 02  750-002983  AK6894      1x OC-12 ATM, SMIR
```

```
 PIC 2        REV 04  750-001894  HD4968      1x G/E, 1000Base-SX
 MMB 1        REV 03  710-001723  HE7264      MMB-144mbit
 ICBM         REV 01  710-003384  HE3042
 PPB 0        REV 01  710-003758  HE7173      PPB Type 2
 PPB 1        REV 01  710-003758  HE7170      PPB Type 2
FPC 5         REV 03  710-001721  HD5001      FPC Type 3
 CPU         REV 06  710-001726          HA5080
 PIC 0        REV 01  750-004535  HC0273      1x OC-192 SM SR1
 PIC 1        REV 01  750-004535  HC0271      1x OC-192 SM SR1
 PIC 2        REV 01  750-004535  HC0254      1x OC-192 SM SR1
 MMB 0        REV 03  710-001723  HE7263      MMB-144mbit
 MMB 1        REV 03  710-001723  HE7266      MMB-144mbit
 ICBM         REV 01  710-003384  HE3044
 PPB 0        REV 02  710-002845  HD6027      PPB Type 3
 PPB 1        REV 02  710-002845  HD6039      PPB Type 3
FPC 7         REV 01  710-002385  HE3179      FPC Type 2
 CPU         REV 06  710-001726          HE7915
 PIC 0        REV 01  710-002381  AD3632      2x G/E, 1000 BASE-SX
 PIC 1        REV 05  750-001901  AD3831      4x OC-12 SONET, SMIR
 PIC 2        REV 01  750-001900  AA9603      1x OC-48 SONET, SMIR
 PIC 3        REV 05  750-001900  AD5724      1x OC-48 SONET, SMSR
 MMB 1        REV 02  710-004047  HE3424      MMB-288mbit
 ICBM         REV 04  710-003384  HA4480
 PPB 0        REV 02  710-003758  HE3169      PPB Type 2
 PPB 1        REV 02  710-003758  HA4535      PPB Type 2
SPMB 0       REV 02  710-003229 HF6876
SPMB 1       REV 02  710-003229 HG6237
SIB 0         REV 02  710-005157  HJ9669      SIB-I8-F16
SIB 1         REV 02  710-005157  HJ9668      SIB-I8-F16
SIB 2         REV 02  710-005157  HH3039      SIB-I8-F16
SIB 3         REV 02  710-005157  HH3041      SIB-I8-F16
SIB 4         REV 02  710-005157  HJ9657      SIB-I8-F16
```

Table 47 on page 258 provides an example of **jnxContentEntry** objects in the **jnxContentTable** of a T320 router.

Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	**jnxMidplaneT320.0**	Midplane	AY4527	Rev 01	(0) 0:00:00.00	710-004339
2	1	0	0	**jnxT320Power.0**	PEM 0	ML14099	Rev 01	(0) 0:00:00.00	–
4	1	1	0	**jnxT320Fan.0**	Top left front fan	–	–	(0) 0:00:00.00	–
4	1	2	0	**jnxT320Fan.0**	Top left middle fan	–	–	(0) 0:00:00.00	–
4	1	3	0	**jnxT320Fan.0**	Top left rear fan	–	–	(0) 0:00:00.00	–

Copyright © 2012, Juniper Networks, Inc.

Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	1	4	0	**jnxT320Fan.0**	Top right front fan	–	–	(0) 0:00:00.00	–
4	1	5	0	**jnxT320Fan.0**	Top right middle fan	–	–	(0) 0:00:00.00	–
4	1	6	0	**jnxT320Fan.0**	Top right rear fan	–	–	(0) 0:00:00.00	–
4	2	1	0	**jnxT320Fan.0**	Bottom left front fan	–	–	(0) 0:00:00.00	–
4	2	2	0	**jnxT320Fan.0**	Bottom left middle fan	–	–	(0) 0:00:00.00	–
4	2	3	0	**jnxT320Fan.0**	Bottom left rear fan	–	–	(0) 0:00:00.00	–
4	2	4	0	**jnxT320Fan.0**	Bottom right front fan	–	–	(0) 0:00:00.00	–
4	2	5	0	**jnxT320Fan.0**	Bottom right middle fan	–	–	(0) 0:00:00.00	–
4	2	6	0	**jnxT320Fan.0**	Bottom right rear fan	–	–	(0) 0:00:00.00	–
4	3	1	0	**jnxT320Fan.0**	Rear tray top fan	–	–	(0) 0:00:00.00	–
4	3	2	0	**jnxT320Fan.0**	Rear tray second fan	–	–	(0) 0:00:00.00	–
4	3	3	0	**jnxT320Fan.0**	Rear tray middle fan	–	–	(0) 0:00:00.00	–
4	3	4	0	**jnxT320Fan.0**	Rear tray fourth fan	–	–	(0) 0:00:00.00	–
4	3	5	0	**jnxT320Fan.0**	Rear tray bottom fan	–	–	(0) 0:00:00.00	–
7	4	0	0	**jnxT320FPC.0**	FPC @ 3/*/*	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333

Copyright © 2012, Juniper Networks, Inc.

Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	4	1	0	**jnxT320FPC.0**	FPC @ 3/0/* top temperature sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	2	0	**jnxT320FPC.0**	FPC @ 3/1/* bottom temperature sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
8	1	1	0	**jnxT320PIC3**	PIC: 1x OC-192 SM SR2 @ 0/0/*	HJ9283	REV 06	(6378) 0:01:03.78	750-004535
8	1	2	0	**jnxT320PIC3**	PIC: 1x OC-192 SM SR2 @ 0/1/*	HJ9298	REV 06	(6434) 0:01:04.34	750-004535
9	1	0	0	**jnxT320HM.0**	Host 0	2108 6570 0286	REV 01	(32762924) 3 days, 19:00:29.24	740-005022
9	2	0	0	**jnxT320HM.0**	Host 1	2109 2900 0186	REV 01	(110269900) 12 days, 18:18:19.00	740-005022
10	1	0	0	**jnxT320FPB.0**	FPM	AY4514	REV 02	(0) 0:00:00.00	710-004461
11	1	0	0	**jnxT320SCG.0**	SCG 0	AY4520	REV 06	(0) 0:00:00.00	710-004455
11	2	0	0	**jnxT320SCG.0**	SCG 1	AY4526	REV 06	(0) 0:00:00.00	710-004455
12	1	0	0	**jnxT320CB.0**	CB 0	AY4765	REV 11	(0) 0:00:00.00	710-002728
12	2	0	0	**jnxT320CB.0**	CB 1	HG6051	REV 06	(0) 0:00:00.00	710-002728
13	1	0	0	**jnxT320CIP.0**	CIP	HC0476	REV 05	(0) 0:00:00.00	710-002895
14	1	0	0	**jnxT320SPMB.0**	SPMB 0	HB1893	REV 02	(26186997) 3 days, 0:44:29.97	710-003229

Copyright © 2012, Juniper Networks, Inc.

Table 47: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
14	2	0	0	**jnxT320SPMB.0**	SPMB 1	HD5520	REV 02	(26186913) 3 days, 0:44:29.13	710-003229
15	1	0	0	**jnxT320SIB.0**	SIB 0	BC1509	REV 02	(0) 0:00:00.00	710-005157
15	2	0	0	**jnxT320SIB.0**	SIB 1	BC1512	REV 02	(0) 0:00:00.00	710-005157
15	3	0	0	**jnxT320SIB.0**	SIB 2	BC1494	REV 02	(0) 0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T320 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis  T320
Midplane         REV 01   710-004339   AY4527
FPM GBUS         REV 02   710-004461   AY4514
FPM Display      REV 02   710-002897   HF6097
CIP              REV 05   710-002895   HC0476
PEM 0            Rev 01   740-004359   ML14099           Power Entry
Module
SCG 0            REV 06   710-004455   AY4520
SCG 1            REV 06   710-004455   AY4526
RE 0               REV 01   740-005022   210865700286      RE-3.0
RE 1               REV 01   740-005022   210929000186      RE-3.0
CB 0             REV 11   710-002728   AY4765
CB 1             REV 06   710-002728   HG6051
FPC 1            REV 01   710-004333   AY4507            FPC Type 3
  CPU            REV 06   710-001726   HA4719
  MMB 1          REV 03   710-004047   HD5738            MMB-288mbit
  PPB 0          REV 02   710-002845   HC0988            PPB Type 3
FPC 3            REV 01   710-004333   AY4706            FPC Type 3
  CPU            REV 06   710-001726   HE7916
  MMB 1          REV 03   710-004047   HG6326            MMB-288mbit
  PPB 0          REV 02   710-002845   HC0958            PPB Type 3
SPMB 0           REV 02   710-003229   HB1893
SPMB 1           REV 02   710-003229   HD5520
SIB 0            REV 02   710-005157   BC1509            SIB-I8-F16
SIB 1            REV 02   710-005157   BC1512            SIB-I8-F16
SIB 2            REV 02   710-005157   BC1494            SIB-I8-F16
```

Table 48 on page 262 provides an example of **jnxContentstEntry** objects in the **jnxContentTable** of a T4000 router.

Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T4000 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	**jnxMidplaneT1600.0**	Midplane	SNRC8652	REV 01	0:0:00:00.00	710-027486
2	1	0	0	**jnxPower.0**	PEM 0	MD21815	REV 01	0:0:00:00.00	40-026384
4	1	0	0	**jnxT4000Fan.0**	Front Top Fan Tray	–	–	0:0:00:00.00	–
4	1	1	0	**jnxT4000Fan.0**	Top Left Front fan	–	–	0:0:00:00.00	–
4	1	2	0	**jnxT4000Fan.0**	Top Left Middle fan	–	–	0:0:00:00.00	–
4	1	3	0	**jnxT4000Fan.0**	Top Left Rear fan	–	–	0:0:00:00.00	–
4	1	4	0	**jnxT4000Fan.0**	Top Right Front fan	–	–	0:0:00:00.00	–
4	1	5	0	**jnxT4000Fan.0**	Top right Middle fan	–	–	0:0:00:00.00	–
4	1	6	0	**jnxT4000Fan.0**	Top Right Rear Fan	–	–	0:0:00:00.00	–
4	2	0	0	**jnxT4000Fan.0**	Front Bottom fanTray	–	–	0:0:00:00.00	–
4	2	1	0	**jnxT4000Fan.0**	BottomLeft Front fan	–	–	0:0:00:00.00	–
4	2	2	0	**jnxT4000Fan.0**	BottomLeft Middle Fan	–	–	0:0:00:00.00	–
4	2	3	0	**jnxT4000Fan.0**	BottomLeft Rear fan	–	–	0:0:00:00.00	–
4	2	4	0	**jnxT4000Fan.0**	Bottom Right Front Fan	–	–	0:0:00:00.00	–
4	2	5	0	**jnxT4000Fan.0**	Bottom Right Middle Fan	–	–	0:0:00:00.00	–
4	2	6	0	**jnxT4000Fan.0**	Bottom Right Rear Fan	–	–	0:0:00:00.00	–
4	3	0	0	**jnxT4000Fan.0**	Rear fan Tray	–	–	0:0:00:00.00	–

Copyright © 2012, Juniper Networks, Inc.

Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T4000 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	3	1	0	**jnxT4000Fan.0**	Rear Tray Top Fan	–	–	0:0:00:00.00	–
4	3	2	0	**jnxT4000Fan.0**	Rear Tray Second Fan	–	–	0:0:00:00.00	–
4	3	3	0	**jnxT4000Fan.0**	Rear Tray Third Fan	–	–	0:0:00:00.00	–
4	3	4	0	**jnxT4000Fan.0**	Rear Tray Fourth Fan	–	–	0:0:00:00.00	–
4	3	5	0	**jnxT4000Fan.0**	Rear Tray Fifth Fan	–	–	0:0:00:00.00	–
4	3	6	0	**jnxT4000Fan.0**	Rear Tray Sixth Fan	–	–	0:0:00:00.00	–
4	3	7	0	**jnxT4000Fan.0**	Rear Tray Seventh Fan	–	–	0:0:00:00.00	–
4	3	8	0	**jnxT4000Fan.0**	Rear Tray Bottom Fan	–	–	0:0:00:00.00	–
7	2	0	0	**jnxT4000FPC.0**	FPC Type 4-ES@ 1/*/*	SNBBA3691	REV 03	0:18:57:02.71	710-033871
7	2	1	0	**jnxT4000FPC.0**	FPC Type 4-ES@ 1/*/* top temperature sensor	SNBBA3691	REV 03	0:18:57:02.71	710-033871
7	2	2	0	**jnxT4000FPC.0**	FPC Type 4-ES@ 1/*/* bottom temperature sensor	SNBBA3691	REV 03	0:18:57:02.71	710-033871
7	3	0	0	**jnxT4000FPC.0**	FPC Type 5-3D@ 2/*/*	SNEG3636	REV 07	0:18:56:52.85	750-010153
7	3	1	0	**jnxT4000FPC.0**	FPC Type 5-3D@ 2/*/*	SNEG3636	REV 07	0:18:56:52.85	750-010153
7	3	2	0	**jnxT4000FPC.0**	FPC Type 5-3D@ 2/*/* exhaust	SNEG3636	REV 07	0:18:56:52.85	750-010153

Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T4000 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	5	0	0	**jnxT4000FPC.0**	FPC Type 5-3D @ 4/*/*	SNEG3641	REV 07	0:18:56:52.85	750-032819
7	5	1	0	**jnxT4000FPC.0**	FPC Type 5-3D @ 4/*/* intake	SNEG3641	REV 07	0:18:56:52.85	750-032819
7	5	2	0	**jnxT4000FPC.0**	FPC Type 5-3D @ 4/*/* exhaust	SNEG3641	REV 07	0:18:56:52.85	750-032819
8	3	1	0	**jnxT4000PIC3.0**	PIC: 12x10GE (LAN/WAN) SFPP @ 2/0/*	SNEE3734	REV 02	0:18:57:04.47	50-034624
8	3	2	0	**jnxT4000PIC3.0**	PIC: 12x10GE (LAN/WAN) SFPP @ 2/1/*	SNEE3705	REV 02	0:18:57:04.55	750-034624
8	5	1	0	**jnxT4000PIC3.0**	PIC: 12x10GE (LAN/WAN) SFPP @ 4/1/*	SNEE3708	REV 02	0:18:56:55.16	750-034624
9	1	2	0	**jnxT4000HM.0**	Routing Engine 0	P37F02865	REV 06	0:19:19:30.95	740-02694
9	2	0	0	**jnxT4000HM.0**	Routing Engine 1	P37F02853	REV 06	2:19:45:51.00	740-026941
10	1	1	0	**jnxT4000FPB.0**	FPM GBUS	SNBBA7038	REV 13	0:0:00:00.00	710-002901
10	1	2	0	**jnxT4000FPB.0**	FPM Display	SNBBA7038	REV 13	0:0:00:00.00	710-021387
11	1	0	0	**jnxT4000SCG.0**	SCG 0	SNBBA5661	REV 18	0:0:00:00.00	710-003423
11	2	0	0	**jnxT4000SCG.0**	SCG 1	SNBBA5661	REV 18	0:0:00:00.00	710-003423
12	1	0	0	**jnxT4000CB.0**	CB 0	SNEF7357	REV 09	0:0:00:00.00	710-022597
12	2	0	0	**jnxT4000CB.0**	CB 1	SNEF7368	REV 09	0:0:00:00.00	710-022597
13	1	0	0	**jnxT4000CIP.0**	CIP	SNBBA5206	REV 06	0:0:00:00.00	710-002895
14	1	0	0	**jnxT4000SPMB.0**	SPMB 0	SNEF6110	REV 05	0:18:56:06.72	710-023321
14	2	0	0	**jnxT4000SPMB.0**	SPMB 1	SNEF6100	REV 05	0:18:56:08.01	710-023321
15	1	0	0	**jnxT4000SIB.0**	SIB 0	SNEE6281	REV 02	0:0:00:00.00	710-023321

Copyright © 2012, Juniper Networks, Inc.

Table 48: jnxContentsEntry Objects in the jnxContentsTable of a T4000 Router *(continued)*

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
15	2	0	0	**jnxT4000SIB.0**	SIB 1	SNEG2262	REV 07	0:0:00:00.00	711-036340
15	3	0	0	**jnxT4000SIB.0**	SIB 2	SNEF3308	REV 02	0:0:00:00.00	711-036340
15	4	0	0	**jnxT4000SIB.0**	SIB 3	SNEF1317	REV 02	0:0:00:00.00	711-036340
15	5	0	0	**jnxT4000SIB.0**	SIB 4	SNEG2301	REV 07	0:0:00:00.00	711-036340

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T4000 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JN1172F25AHA      T4000
Midplane         REV 01   710-027486   RC8355            T-series Backplane
FPM GBUS         REV 13   710-002901   BBAE0927          T640 FPM Board
FPM Display      REV 01   710-021387   EF6764            T1600 FPM Display
CIP              REV 06   710-002895   BBAD9210          T-series CIP
PEM 0            REV 01   740-036442   VA00016           Power Entry Module 6x60
SCG 0            REV 18   710-003423   BBAD7248          T640 Sonet Clock Gen.
SCG 1            REV 18   710-003423   BBAE3874          T640 Sonet Clock Gen.
Routing Engine 0 REV 05   740-026941   P737F-002248      RE-DUO-1800
Routing Engine 1 REV 06   740-026941   P737F-002653      RE-DUO-1800
CB 0             REV 09   710-022597   ED0295            LCC Control Board
CB 1             REV 09   710-022597   EA6050            LCC Control Board
FPC 0            REV 26   750-032819   EK1173            FPC Type 5-3D
  CPU            REV 12   711-030686   EJ8584            SNG PMB
  PIC 0          REV 07   750-034624   EF6837            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   123363A01145      SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   123363A01147      SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01P3           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B10M03256         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   AJJ01M2           SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   123363A01137      SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01PN           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   AJJ01NW           SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   123363A01139      SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJJ01KE           SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   123363A01336      SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B10M01325         SFP+-10G-SR
  PIC 1          REV 07   750-034624   EF6800            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   AJJ01SA           SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   AJJ01QZ           SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJH0217           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   AJJ01TE           SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   AJJ01KV           SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   AJJ01MU           SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01R0           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   AJJ01TC           SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   AJJ0364           SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJD0GV3           SFP+-10G-SR
```

```
    Xcvr 10      REV 01   740-031980   B10M03343         SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   AJJ01QJ           SFP+-10G-SR
  LMB 0          REV 05   711-034381   EJ8490            Type-0 LMB
  LMB 1          REV 04   711-035774   EJ8517            Type-1 LMB
  LMB 2          REV 05   711-034381   EJ8489            Type-0 LMB
FPC 3            REV 07   750-032819   EG3637            FPC Type 5-3D
  CPU            REV 09   711-030686   EG0150            SNG PMB
  PIC 0          REV 08   750-035293   EF3657            1x100GE
    Xcvr 0       REV 01   740-032210   C22CQNJ           CFP-100G-LR4
  PIC 1          REV 10   750-034624   BBAN4098          12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   B11J04902         SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   B11J04891         SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01MX           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B11J04183         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B11J04894         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   B11J04184         SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   B11J04897         SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   B11J04899         SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   AJJ01TV           SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   B11J04057         SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   AJJ01M4           SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B11J04905         SFP+-10G-SR
  LMB 0          REV 04   711-034381   EG1524            Type-0 LMB
  LMB 1          REV 03   711-035774   EG0345            Type-1 LMB
  LMB 2          REV 04   711-034381   EG1522            Type-0 LMB
FPC 5            REV 03   710-033871   BBAJ0768          FPC Type 4-ES
  CPU            REV 11   710-016744   BBAH9342          ST-PMB2
  PIC 0          REV 09   750-029262   EE6789            100GE
  PIC 1          REV 03   750-034781   EE6655            100GE CFP
    Xcvr 0       REV 01   740-032210   J11A22334         CFP-100G-LR4
  BRIDGE 0       REV 03   711-029995   EE6572            100GE Bridge Board
  MMB 0          REV 07   710-025563   BBAJ4657          ST-MMB2
  MMB 1          REV 07   710-025563   BBAJ3073          ST-MMB2
FPC 6            REV 05   750-010153   EF4936            FPC Type 5-3D
  CPU            REV 06   711-030686   EF4189            SNG PMB
  PIC 0          REV 10   750-034624   BBAN4109          12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   B11J04895         SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   B11J04898         SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   B11J04021         SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B11J04903         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B11J04311         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   B11J04059         SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   B11J04016         SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   B11J04017         SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   B11J04887         SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   B11J04297         SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   B11J04893         SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B11J04022         SFP+-10G-SR
  PIC 1          REV 02   750-034624   EE3711            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   AJH033X           SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   AJJ01N0           SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01SV           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   AJJ032L           SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B10M01593         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   AJD0FF1           SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01NU           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   123363A01305      SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   B10M00361         SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJJ01M7           SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   AJJ032X           SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   AJJ01PG           SFP+-10G-SR
```


Copyright © 2012, Juniper Networks, Inc.

```
  LMB 0          REV 04   711-034381   EF3838            Type-0 LMB
  LMB 1          REV 03   711-035774   EF3821            Type-1 LMB
  LMB 2          REV 04   711-034381   EF3834            Type-0 LMB
SPMB 0           REV 05   710-023321   ED1990            LCC Switch CPU
SPMB 1           REV 05   710-023321   EA2768            LCC Switch CPU
SIB 0            REV 02   711-036340   EF8802            SIB-HC-3D
SIB 1            REV 07   711-036340   EG2286            SIB-HC-3D
SIB 2            REV 07   711-036340   EG2252            SIB-HC-3D
SIB 3            REV 02   711-036340   EF1358            SIB-HC-3D
SIB 4            REV 02   711-036340   EF8806            SIB-HC-3D
Fan Tray 0                                               Front Top Fan Tray
Fan Tray 1                                               Front Bottom Fan Tray
-- Rev 2
Fan Tray 2                                               Rear Fan Tray -- Rev 3
```

jnxLEDLastChange

The object identifier for the **jnxLEDLastChange** object is **{jnxBoxAnatomy 9}**. This object indicates when the LED last changed state. Its value is 0 if the **sysUpTime** value is unknown or if it already existed when the agent was active.

jnxLEDTable

The object identifier for the **jnxLEDTable** object is **{jnxBoxAnatomy 10}**. This object indicates the LED status of the router and lists the contents of an entry. Entries in the **jnxLEDTable** are represented by the **jnxLEDEntry** object, whose object identifier is **{jnxLEDTable 1}**.

The **jnxLEDTable** describes the components of the LED Box Indicators, whose elements are described as follows:

- **jnxLEDAssociateTable**—The associate table to which the entry is related, whose object identifier is **{jnxLEDEntry 1}**.
- **jnxLEDAssociateIndex**—The index of the subject in the associated table to which the entry is related, whose object identifier is **{jnxLEDEntry 2}**. The associate index is the index of the subject in the associated table, which returns you to the **jnxContainersTable**.
- **jnxLEDL1Index**—The level-one index of the associate table to which an entry is related, whose object identifier is **{jnxLEDEntry 3}**. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.

  NOTE: MIBs start with a value of 1, while the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- **jnxLEDL2Index**—The level-two index of the associate table to which an entry is related, whose object identifier is **{jnxLEDEntry 4}**. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.

- **jnxLEDL3Index**—The level-three index of the associate table to which an entry is related, whose object identifier is **{jnxLEDEntry 5}**. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- **jnxLEDOriginator**—The chassis component that originated the update, whose object identifier is **{jnxLEDEntry 6}**.
- **jnxLEDDescr**—The name or detailed description of the entry, whose object identifier is **{jnxLEDEntry 7}**.
- **jnxLEDState**—The state of the LED indicator, whose object identifier is **{jnxLEDEntry 8}**. The state is represented by one of the following values:
  - Amber—Alarm, offline, not working
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Other—Unknown or unavailable
  - Red—Alert, component failed
  - Yellow—Alarm, warning
- **jnxLEDStateOrdered**—The state of the LED indicator, whose object identifier is **{jnxLEDEntry 9}**. **jnxLEDStateOrdered** provides the same information as **jnxLEDState** but lists the states in a different order. The states are defined as follows:
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Amber—Alarm, offline, not working
  - Yellow—Alarm, warning
  - Red—Alert, component failed
  - Other—Unknown or unavailable

Table 49 on page 269 through Table 51 on page 270 provide examples of **jnxLEDEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxLEDEntry** objects:

- Associate table—**jnxLEDAssociateTable**
- Associate index—**jnxLEDAssociateIndex**
- L1 Index—**jnxLEDL1Index**
- L2 Index—**jnxLEDL2Index**
- L3 Index—**jnxLEDL3Index**
- Originator—**jnxLEDOriginator**

 Copyright © 2012, Juniper Networks, Inc.

- Description—**jnxLEDDescr**
- State—**jnxLEDState**

Table 49 on page 269 provides an example of **jnxLEDEntry** objects in the **jnxLEDTable** of an M20 router.

Table 49: jnxLEDEntry Objects in the jnxLEDTable of an M20 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	**jnxChassisM20.0**	Chassis alarm LED	Other
jnxContentsTable	6	1	0	0	**jnxM20SSB.0**	SSB 1 LED	Blue
jnxContentsTable	6	2	0	0	**jnxM20SSB.0**	SSB 2 LED	Green
jnxContentsTable	7	1	0	0	**jnxM20FPC.0**	FPC 1 LED	Amber
jnxContentsTable	7	2	0	0	**jnxM20FPC.0**	FPC 2 LED	Blue
jnxContentsTable	7	3	0	0	**jnxM20FPC.0**	FPC 3 LED	Blue
jnxContentsTable	7	4	0	0	**jnxM20FPC.0**	FPC 4 LED	Amber
jnxContentsTable	9	1	0	0	**jnxM20RE.0**	Routing Engine 1 LED	Blue
jnxContentsTable	9	2	0	0	**jnxM20RE.0**	Routing Engine 2 LED	Other

Table 50 on page 269 provides an example of **jnxLEDEntry** objects in the **jnxLEDTable** of a T640 router.

Table 50: jnxLEDEntry Objects in the jnxLEDTable of a T640 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	**jnxChassisT640.0**	Chassis alarm LED	Other
jnxContentsTable	7	1	0	0	**jnxT640FPC.0**	FPC slot 0 LED	Other
jnxContentsTable	7	2	0	0	**jnxT640FPC.0**	FPC slot 1 LED	Green
jnxContentsTable	7	3	0	0	**jnxT640FPC.0**	FPC slot 2 LED	Other
jnxContentsTable	7	4	0	0	**jnxT640FPC.0**	FPC slot 3 LED	Other
jnxContentsTable	7	5	0	0	**jnxT640FPC.0**	FPC slot 4 LED	Other
jnxContentsTable	7	6	0	0	**jnxT640FPC.0**	FPC slot 5 LED	Green

Copyright © 2012, Juniper Networks, Inc.

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

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	7	7	0	0	**jnxT640FPC.0**	FPC slot 6 LED	Other
jnxContentsTable	7	8	0	0	**jnxT640FPC.0**	FPC slot 7 LED	Green
jnxContentsTable	9	1	0	0	**jnxT640HM.0**	Host 0 LED	Blue
jnxContentsTable	9	2	0	0	**jnxT640HM.0**	Host 1 LED	Green

Table 51 on page 270 provides an example of **jnxLEDEntry** objects in the **jnxLEDTable** of a T320 router.

Table 51: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable(3)	1	1	0	0	**jnxChassisT320.0**	Chassis alarm LED	Other
jnxContentsTable(3)	7	1	0	0	**jnxT320FPC.0**	FPC slot 0 LED	Other
jnxContentsTable(3)	7	2	0	0	**jnxT320FPC.0**	FPC slot 1 LED	Other
jnxContentsTable(3)	7	3	0	0	**jnxT320FPC.0**	FPC slot 2 LED	Other
jnxContentsTable(3)	7	4	0	0	**jnxT320FPC.0**	FPC slot 3 LED	Other
jnxContentsTable(3)	7	5	0	0	**jnxT320FPC.0**	FPC slot 4 LED	Other
jnxContentsTable(3)	7	6	0	0	**jnxT320FPC.0**	FPC slot 5 LED	Other
jnxContentsTable(3)	7	7	0	0	**jnxT320FPC.0**	FPC slot 6 LED	Other
jnxContentsTable(3)	7	8	0	0	**jnxT320FPC.0**	FPC slot 7 LED	Other
jnxContentsTable(3)	9	1	0	0	**jnxT320HM.0**	Host 0 LED	Blue
jnxContentsTable(3)	9	2	0	0	**jnxT320HM.0**	Host 1 LED	Green

jnxFilledLastChange

The object identifier for the **jnxFilledLastChange** object is **{jnxBoxAnatomy 11}**. This object indicates when the box filled status last changed. This variable is 0 if the **sysUpTime** value is unknown or it already existed when the agent was active.

Copyright © 2012, Juniper Networks, Inc.

jnxFilledTable

The object identifier for the **jnxFilledTable** object is **{jnxBoxAnatomy 12}**. This object indicates whether a specific container in the router is used (filled) or empty. This table is used for inventory and capacity planning.

Entries in the **jnxFilledTable** are represented by the **jnxFilledEntry** object, whose object identifier is **{jnxFilledTable 1}**.

The **jnxFilledTable** describes the status of specific containers whose component objects are described as follows:

- **jnxFilledContainerIndex**—The associated **jnxContainersIndex** in the **jnxContainersTable**, whose object identifier is **{jnxFilledEntry 1}**.
- **jnxFilledL1Index**—The level-one index of the container housing the entry, whose object identifier is **{jnxFilledEntry 2}**.
- **jnxFilledL2Index**—The level-two index of the container housing the entry, whose object identifier is **{jnxFilledEntry 3}**.
- **jnxFilledL3Index**—The level-three index of the container housing the entry, whose object identifier is **{jnxFilledEntry 4}**.
- **jnxFilledDescr**—The entry's name or detailed description of the entry, whose object identifier is **{jnxFilledEntry 5}**.
- **jnxFilledState**—The entry's state (filled or empty), whose object identifier is **{jnxFilledEntry 6}**.

Table 52 on page 271 through Table 54 on page 278 provide examples of **jnxFilledEntry** objects in the **jnxFilledTable**. The following column headings for each table are abbreviated to correspond to the parts of the **jnxFilledEntry** objects:

- Container index—**jnxFilledContainerIndex**
- L1—**jnxFilledL1Index**
- L2—**jnxFilledL2Index**
- L3—**jnxFilledL3Index**
- Description—**jnxFilledDescr**
- State—**jnxFilledState**

Table 52 on page 271 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of an M20 router.

Table 52: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame compartment	Filled

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
1	1	1	0	Temperature sensor space 0	Filled
1	1	2	0	Temperature sensor space 1	Filled
2	1	0	0	Power supply compartment A	Filled
2	2	0	0	Power supply compartment B	Empty
3	1	0	0	Rear top impeller compartment	Filled
3	2	0	0	Front bottom impeller compartment	Filled
4	1	0	0	Rear left fan compartment	Filled
4	2	0	0	Right center fan compartment	Filled
4	3	0	0	Rear right fan compartment	Filled
5	1	0	0	Host controller compartment	Filled
6	1	0	0	SCB slot	Filled
7	1	0	0	FPC slot 0	Empty
7	2	0	0	FPC slot 1	Empty
7	3	0	0	FPC slot 2	Filled
7	4	0	0	FPC slot 3	Filled
7	5	0	0	FPC slot 4	Empty
7	6	0	0	FPC slot 5	Filled
7	7	0	0	FPC slot 6	Empty
7	8	0	0	FPC slot 7	Empty
8	1	1	0	PIC space @ 0/0/*	Empty
8	1	2	0	PIC space @ 0/1/*	Empty
8	1	3	0	PIC space @ 0/2/*	Empty
8	1	4	0	PIC space @ 0/3/*	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
8	2	1	0	PIC space @ 1/0/*	Empty
8	2	2	0	PIC space @ 1/1/*	Empty
8	2	3	0	PIC space @ 1/2/*	Empty
8	2	4	0	PIC space @ 1/3/*	Empty
8	3	1	0	PIC space @ 2/0/*	Filled
8	3	2	0	PIC space @ 2/1/*	Filled
8	3	3	0	PIC space @ 2/2/*	Filled
8	3	4	0	PIC space @ 2/3/*	Filled
8	4	1	0	PIC space @ 3/0/*	Filled
8	4	2	0	PIC space @ 3/1/*	Filled
8	4	3	0	PIC space @ 3/2/*	Filled
8	4	4	0	PIC space @ 3/3/*	Filled
8	5	1	0	PIC space @ 4/0/*	Empty
8	5	2	0	PIC space @ 4/1/*	Empty
8	5	3	0	PIC space @ 4/2/*	Empty
8	5	4	0	PIC space @ 4/3/*	Empty
8	6	1	0	PIC space @ 5/0/*	Filled
8	6	2	0	PIC space @ 5/1/*	Filled
8	6	3	0	PIC space @ 5/2/*	Filled
8	6	4	0	PIC space @ 5/3/*	Filled
8	7	1	0	PIC space @ 6/0/*	Empty
8	7	2	0	PIC space @ 6/1/*	Empty
8	7	3	0	PIC space @ 6/2/*	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
8	7	4	0	PIC space @ 6/3/*	Empty
8	8	1	0	PIC space @ 7/0/*	Empty
8	8	2	0	PIC space @ 7/1/*	Empty
8	8	3	0	PIC space @ 7/2/*	Empty
8	8	4	0	PIC space @ 7/3/*	Empty
9	1	0	0	Routing Engine compartment	Filled

Table 53 on page 274 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T640 router.

Table 53: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Empty
2	2	0	0	PEM slot 1	Filled
4	1	1	0	Top left front fan slot	Filled
4	1	2	0	Top left middle fan slot	Filled
4	1	3	0	Top left rear fan slot	Filled
4	1	4	0	Top right front fan slot	Filled
4	1	5	0	Top right middle fan slot	Filled
4	1	6	0	Top right rear fan slot	Filled
4	2	1	0	Bottom left front fan slot	Filled
4	2	2	0	Bottom left middle fan slot	Filled
4	2	3	0	Bottom left rear fan slot	Filled
4	2	4	0	Bottom right front fan slot	Filled

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
4	2	5	0	Bottom right middle fan slot	Filled
4	2	6	0	Bottom right rear fan slot	Filled
4	3	1	0	Fourth blower from top slot	Filled
4	3	2	0	Bottom blower slot	Filled
4	3	3	0	Middle blower slot	Filled
4	3	4	0	Top blower slot	Filled
4	3	5	0	Second blower from top slot	Filled
7	3	2	0	FPC slot 0	Empty
7	3	3	0	FPC slot 0 top temperature sensor	Empty
7	3	4	0	FPC slot 0 bottom temperature sensor	Empty
7	3	5	0	FPC slot 1	Filled
7	3	6	0	FPC slot 1 top temperature sensor	Filled
7	1	0	0	FPC slot 1 bottom temperature sensor	Filled
7	1	1	0	FPC slot 2	Empty
7	1	2	0	FPC slot 2 top temperature sensor	Empty
7	2	0	0	FPC slot 2 bottom temperature sensor	Empty
7	2	1	0	FPC slot 3	Empty
7	2	2	0	FPC slot 3 top temperature sensor	Empty
7	3	0	0	FPC slot 3 bottom temperature sensor	Empty
7	3	1	0	FPC slot 4	Empty
7	3	2	0	FPC slot 4 top temperature sensor	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
7	4	0	0	FPC slot 4 bottom temperature sensor	Empty
7	4	1	0	FPC slot 5	Filled
7	4	2	0	FPC slot 5 top temperature sensor	Filled
7	5	0	0	FPC slot 5 bottom temperature sensor	Filled
7	5	1	0	FPC slot 6	Empty
7	5	2	0	FPC slot 6 top temperature sensor	Empty
7	6	0	0	FPC slot 6 bottom temperature sensor	Empty
7	6	1	0	FPC slot 7	Filled
7	6	2	0	FPC slot 7 top temperature sensor	Filled
7	7	0	0	FPC slot 7 bottom temperature sensor	Filled
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	1	3	0	PIC slot @ 0/2/*	Empty
8	1	4	0	PIC slot @ 0/3/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Filled
8	2	2	0	PIC slot @ 1/1/*	Filled
8	2	3	0	PIC slot @ 1/2/*	Filled
8	2	4	0	PIC slot @ 1/3/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	3	3	0	PIC slot @ 2/2/*	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
8	3	4	0	PIC slot @ 2/3/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Empty
8	4	2	0	PIC slot @ 3/1/*	Empty
8	4	3	0	PIC slot @ 3/2/*	Empty
8	4	4	0	PIC slot @ 3/3/*	Empty
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	3	0	PIC slot @ 4/2/*	Empty
8	5	4	0	PIC slot @ 4/3/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Filled
8	6	2	0	PIC slot @ 5/1/*	Filled
8	6	3	0	PIC slot @ 5/2/*	Filled
8	6	4	0	PIC slot @ 5/3/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	7	3	0	PIC slot @ 6/2/*	Empty
8	7	4	0	PIC slot @ 6/3/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Filled
8	8	2	0	PIC slot @ 7/1/*	Filled
8	8	3	0	PIC slot @ 7/2/*	Filled
8	8	4	0	PIC slot @ 7/3/*	Filled
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled
15	4	0	0	SIB slot 3	Filled
15	5	0	0	SIB slot 4	Filled

Table 54 on page 278 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T320 router.

Table 54: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Filled
2	2	0	0	PEM slot 1	Empty
4	1	1	0	Top left front fan slot	Filled
4	1	2	0	Top left middle fan slot	Filled
4	1	3	0	Top left rear fan slot	Filled

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
4	1	4	0	Top right front fan slot	Filled
4	1	5	0	Top right middle fan slot	Filled
4	1	6	0	Top right rear fan slot	Filled
4	2	1	0	Bottom left front fan slot	Filled
4	2	2	0	Bottom left middle fan slot	Filled
4	2	3	0	Bottom left rear fan slot	Filled
4	2	4	0	Bottom right front fan slot	Filled
4	2	5	0	Bottom right middle fan slot	Filled
4	2	6	0	Bottom right rear fan slot	Filled
4	3	1	0	Rear tray top fan slot	Filled
4	3	2	0	Rear tray second fan slot	Filled
4	3	3	0	Rear tray middle fan slot	Filled
4	3	4	0	Rear tray fourth fan slot	Filled
4	3	5	0	Rear tray bottom fan slot	Filled
7	1	0	0	FPC slot 0	Empty
7	1	1	0	FPC slot top temperature sensor	Empty
7	1	2	0	FPC slot 0 bottom temperature sensor	Empty
7	2	0	0	FPC slot 1	Empty
7	2	1	0	FPC slot 1 top temperature sensor	Empty
7	2	2	0	FPC slot 1 bottom temperature sensor	Empty
7	3	0	0	FPC slot 2	Empty
7	3	1	0	FPC slot 2 top temperature sensor	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
7	3	2	0	FPC slot 2 bottom temperature sensor	Empty
7	4	0	0	FPC slot 3	Filled
7	4	1	0	FPC slot 3 top temperature sensor	Filled
7	4	2	0	FPC slot 3 bottom temperature sensor	Filled
7	5	1	0	FPC slot 4	Empty
7	5	2	0	FPC slot 4 top temperature sensor	Empty
7	5	0	0	FPC slot 4 bottom temperature sensor	Empty
7	6	1	0	FPC slot 5	Empty
7	6	2	0	FPC slot 5 top temperature sensor	Empty
7	6	0	0	FPC slot 5 bottom temperature sensor	Empty
7	7	1	0	FPC slot 6	Empty
7	7	2	0	FPC slot 6 top temperature sensor	Empty
7	7	0	0	FPC slot 6 bottom temperature sensor	Empty
7	8	1	0	FPC slot 7	Empty
7	8	2	0	FPC slot 7 top temperature sensor	Empty
7	8	0	0	FPC slot 7 bottom temperature sensor	Empty
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Empty
8	2	2	0	PIC slot @ 1/1/*	Empty

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Filled
8	4	2	0	PIC slot @ 3/1/*	Filled
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Empty
8	6	2	0	PIC slot @ 5/1/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Empty
8	8	2	0	PIC slot @ 7/1/*	Empty
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled

Copyright © 2012, Juniper Networks, Inc.

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

Container Index	L1	L2	L3	Description	State
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled

Table 55 on page 282 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T4000 router.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Empty
2	2	0	0	PEM slot 1	Filled
4	1	0	0	Front Top Fan Tray slot	Filled
4	1	1	0	Top Left Front Fan slot	Filled
4	1	2	0	Top Left Middle Fan slot	Filled
4	1	3	0	Top Left Rear Fan slot	Filled
4	1	4	0	Top Right Front fan slot	Filled
4	1	5	0	Top Right Middle fan slot	Filled
4	1	6	0	Top Right Rear Fan slot	Filled
4	2	0	0	Front Bottom Fan slot	Filled
4	2	1	0	Bottom Left Front Fan slot	Filled
4	2	2	0	Bottom Left Middle Fan slot	Filled
4	2	3	0	BottomLeft Rear Fan slot	Filled
4	2	4	0	Bottom Right Front Fan slot	Filled
4	2	5	0	Bottom Right Middle Fan slot	Filled
4	2	6	0	Bottom Right Rear Fan slot	Filled

Copyright © 2012, Juniper Networks, Inc.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router *(continued)*

Container Index	L1	L2	L3	Description	State
4	3	0	0	Rear Fan Tray slot	Filled
4	3	1	0	Rear Tray Top Fan slot	Filled
4	3	2	0	Rear Tray Second Fan slot	Filled
4	3	3	0	Rear Tray Third Fan slot	Empty
4	3	4	0	Rear Tray Fourth Fan slot	Empty
4	3	5	0	Rear Tray Fifth Fan slot	Empty
4	3	6	0	Rear Tray Sixth Fan slot	Empty
4	3	7	0	Rear Tray Seventh Fan slot	Empty
4	3	8	0	Rear Tray Bottom Fan slot	Empty
7	1	3	0	FPC slot 0 MMB0	Filled
7	1	4	0	FPC slot 0 MMB1	Filled
7	2	0	0	FPC slot 1	Empty
7	2	1	0	FPC slot 1 top temperature sensor	Empty
7	2	2	0	FPCslot 1 bottom temperature sensor	Empty
7	2	3	0	FPC slot 1 MMB0	Empty
7	2	4	0	FPC slot 1 MMB1	Empty
7	3	0	0	FPC slot 2	Empty
7	3	1	0	FPC slot 2 top temperature sensor	Empty
7	3	2	0	FPC slot 2 bottom temperature sensor	Empty
7	3	3	0	FPC slot 2 MMB0	Empty
7	3	4	0	FPC slot 2 MMB1	Filled
7	4	0	0	FPC slot 3	Filled

Copyright © 2012, Juniper Networks, Inc.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router *(continued)*

Container Index	L1	L2	L3	Description	State
7	4	1	0	FPC slot 3 top temperature sensor	Filled
7	4	2	0	FPC slot 3 bottom temperature sensor	Filled
7	4	3	0	FPC slot 3 MMB0	Filled
7	4	4	0	FPC slot 3 MMB1	Filled
7	5	0	0	FPC slot 4	Filled
7	5	1	0	FPC slot 4 top temperature sensor	Empty
7	5	2	0	FPC slot 4 bottom temperature sensor	Empty
7	5	3	0	FPC slot 4 MMB0	Empty
7	5	4	0	FPC slot 4 MMB1	Empty
7	5	5	0	FPC slot 5	Empty
7	6	1	0	FPC slot 5 top temperature sensor	Empty
7	6	2	0	FPC slot 5 bottom temperature sensor	Filled
7	6	3	0	FPC slot 5 MMB0	Filled
7	6	4	0	FPC slot 5 MMB1	Filled
7	7	0	0	FPC slot 6	Empty
7	7	1	0	FPC slot 6 top temperature sensor	Filled
7	7	2	0	FPC slot 6 bottom temperature sensor	Filled
7	7	3	0	FPC slot 6 MMB0	Filled
7	7	4	0	FPC slot 6 MMB1	Empty
7	8	0	0	FPC slot 7	Filled
7	8	1	0	FPC slot 7 top temperature sensor	Filled

Copyright © 2012, Juniper Networks, Inc.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router *(continued)*

Container Index	L1	L2	L3	Description	State
7	8	2	0	FPC slot 7 bottom temperature sensor	Filled
7	8	3	0	FPC slot 7 MMB0	Filled
7	8	4	0	FPC slot 7 MMB1	Filled
8	1	4	0	PIC slot @ 0/3/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Filled
8	2	2	0	PIC slot @ 1/1/*	Filled
8	2	3	0	PIC slot @ 1/2/*	Filled
8	2	4	0	PIC slot @ 1/3/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	3	3	0	PIC slot @ 2/2/*	Empty
8	3	4	0	PIC slot @ 2/3/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Empty
8	4	2	0	PIC slot @ 3/1/*	Empty
8	4	3	0	PIC slot @ 3/2/*	Empty
8	4	4	0	PIC slot @ 3/3/*	Empty
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	4	0	PIC slot @ 4/3/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Filled
8	6	2	0	PIC slot @ 5/1/*	Filled
8	6	3	0	PIC slot @ 5/2/*	Filled

Copyright © 2012, Juniper Networks, Inc.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router *(continued)*

Container Index	L1	L2	L3	Description	State
8	6	4	0	PIC slot @ 5/3/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	7	3	0	PIC slot @ 6/2/*	Empty
8	7	4	0	PIC slot @ 6/3/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Filled
8	8	2	0	PIC slot @ 7/1/*	Filled
8	8	3	0	PIC slot @ 7/2/*	Filled
8	8	4	0	PIC slot @ 7/3/*	Filled
9	1	0	0	Routing Engine 0 slot	Filled
9	2	0	0	Routing Engine 1 slot	Filled
10	1	1	0	FPM GBUS slot	Filled
10	1	2	0	FPM Display slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled

Copyright © 2012, Juniper Networks, Inc.

Table 55: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router *(continued)*

Container Index	L1	L2	L3	Description	State
15	4	0	0	SIB slot 3	Filled
15	5	0	0	SIB slot 4	Filled

jnxOperatingTable

The object identifier for the **jnxOperatingTable** is **{jnxBoxAnatomy 13}**. This object reports the operating status of various components such as CPU, buffers, and memory.

Juniper Networks devices implement packet forwarding and routing functions with two separate components, the Packet Forwarding Engine and the Routing Engine, to ensure stability. The clean separation of these two functions permits superior forwarding performance and a highly reliable operating system. Therefore, it is not necessary to monitor CPU, memory, and buffer utilization, as is the case with traditional, monolithic code base routers. The Routing Engine has its own CPU, memory, and buffers—separate from those of the Packet Forwarding Engine. The ASIC-based Packet Forwarding Engine forwards packets on all interfaces at wire speed, eliminating the need to monitor packet buffers being exhausted. As a result, CPU utilization under 2 percent is normal.

Entries in the **jnxOperatingTable** are represented by the **jnxOperatingEntry** object, whose object identifier is **{jnxOperatingTable 1}**.

The **jnxOperatingTable** describes the status of specific objects, which are described as follows:

- **jnxOperatingContents**—The associated **jnxContentsIndex** in the **jnxContentsTable**, whose object identifier is **{jnxOperatingEntry 1}**.
- **jnxOperatingL1Index**—The level-one index of the container housing the entry, whose object identifier is **{jnxOperatingEntry 2}**.
- **jnxOperatingL2Index**—The level-two index of the container housing the entry, whose object identifier is **{jnxOperatingEntry 3}**.
- **jnxOperatingL3Index**—The level-three index of the container housing the entry, whose object identifier is **{jnxOperatingEntry 4}**.
- **jnxOperatingDescr**—The name or detailed description of the entry, whose object identifier is **{jnxOperatingEntry 5}**.
- **jnxOperatingState**—The operating state of the entry, whose object identifier is **{jnxOperatingEntry 6}**. The state can be any of the following:
  - **Unknown(1)**—State of the component is unknown or unavailable
  - **Running(2)**—Up and running as an active primary
  - **Ready(3)**—Ready to run; not running yet

  - **Reset(4)**—Held in reset; not ready yet
  - **RunningAtFullSpeed(5)**—Valid for fans only
  - **Down(6)**—Power supply is down or off
  - **Standby(7)**—Running as a standby backup

- **jnxOperatingTemp**—The entry's temperature, in degrees Celsius (ºC), whose object identifier is **{jnxOperatingEntry 7}**.
- **jnxOperatingCPU**—The CPU utilization percentage of the entry, whose object identifier is **{jnxOperatingEntry 8}**. It is valid for the Control Board, the FPC, and the Routing Engine. It is a 5-second rolling-weighted average calculated every second for each of the CPUs. The value is sent to the Routing Engine every 10 seconds. The value for the Routing Engine is calculated as the average CPU usage over a 5-second period. The chassis process (**chassisd**) computes this value for every 5-second period and returns the latest value from the cache when probed for the value of **jnxOperatingCPU.9.1.0.0**, which maps to the Routing Engine CPU. The Routing Engine is the only object of interest; the rest of the objects are most likely zero because CPUs on those cards are only used for management purposes.

  The following three MIB objects provide for better CPU utilization reporting over SNMP:

  - **jnxOperating1MinAvgCPU**—Indicates the average utilization of CPU during the last minute.
  - **jnxOperating5MinAvgCPU**—Indicates the average utilization of CPU during the last 5-minute period.
  - **jnxOperating15MinAvgCPU**—Indicates the average utilization of CPU during the last 15-minute period.

    All these objects return a zero value if the data is not available or is not applicable.

- **jnxOperatingISR**—The CPU utilization percentage of the entry in relation to the interrupt service routing (ISR), whose object identifier is **{jnxOperatingEntry 9}**.
- **jnxOperatingDRAMSize**—The DRAM size of the entry, in bytes, whose object identifier is **{jnxOperatingEntry 10}**. It is valid for the FPC, Routing Engine, and Control Board.
- **jnxOperatingBuffer**—The buffer pool utilization of the entry (a percentage), whose object identifier is **{jnxOperatingEntry 11}**. It is valid for the FPC and Control Board as a percentage of utilization. Buffers are normally fixed-length memory preallocated for read/write, input/output, or reception/transmission. A measurement against these buffers gives some indication of how busy the system is. The larger the percentage utilization, the busier the system. In terms of absolute numbers, the bigger the buffer size, the better the system can handle bursty traffic patterns.
- **jnxOperatingHeap**—The heap utilization of the entry, whose object identifier is **{jnxOperatingEntry 12}**.
- **jnxOperatingUpTime**—The time interval, in 10-millisecond periods, that the entry has been up and running, whose object identifier is **{jnxOperatingEntry 13}**.

 Copyright © 2012, Juniper Networks, Inc.

- **jnxOperatingLastRestart**—The value of **sysUpTime** when the entry was last restarted, whose object identifier is **{jnxOperatingEntry 14}**.
- **jnxOperatingMemory**—The entry's installed memory size, in megabytes (MB), whose object identifier is **{jnxOperatingEntry 15}**.
- **jnxOperatingStateOrdered**—The operating state of the entry, whose object identifier is **{jnxOperatingEntry 16}**. The state is represented by one of the following values:
  - **Running(1)**—Up and running as an active primary
  - **Standby(2)**—Running as a standby backup
  - **Ready(3)**—Ready to run; not running yet
  - **RunningAtFullSpeed(4)**—Valid for fans only
  - **Reset(5)**—Held in reset; not ready yet
  - **Down(6)**—Power supply is down or off
  - **Unknown(7)**—State of the component is unknown or unavailable

Table 56 on page 290 through Table 58 on page 293 provide examples of **jnxOperatingEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxOperatingEntry** objects:

- Contents index—**jnxOperatingContents**
- L1—**jnxOperatingL1Index**
- L2—**jnxOperatingL2Index**
- L3—**jnxOperatingL3Index**
- Description—**jnxOperatingDescr**
- State—**jnxOperatingState**
- Temp—**jnxOperatingTemp**
- CPU—**jnxOperatingCPU**
- ISR—**jnxOperatingISR**
- DRAM—**jnxOperatingDRAMSize**
- Buffer—**jnxOperatingBuffer**
- Heap—**jnxOperatingHeap**
- UpTime—**jnxOperatingUpTime**
- Last Restart—**jnxOperatingLastRestart**
- Memory—**jnxOperatingMemory**

Table 56 on page 290 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of an M20 router.

Table 56: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	26	0	0	0	0	0	0	0:0:00:00.0	0
2	1	0	0	Power supply A	Running	28	0	0	0	0	0	0	0:0:00:00.0	0
2	2	0	0	Power supply B	Running	29	0	0	0	0	0	0	0:0:00:00.0	0
4	1	0	0	Front top fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	2	0	0	Front middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	3	0	0	Front bottom fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	4	0	0	Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
6	1	0	0	SSB 0	Running	30	0	0	671088	6	0	67038195	0:0:00:35.41	64
7	1	0	0	FPC @ 0/*/*	Running	31	0	0	83886	3	0	67035034	0:0:01:06.91	8
7	2	0	0	FPC @ 1/*/*	Running	33	0	0	83886	4	0	67034422	0:0:01:13.04	8
7	3	0	0	FPC @ 2/*/*	Running	31	0	0	83886	3	0	67033809	0:0:01:19.18	8
9	1	0	0	Routing Engine 0	Running	29	4	0	802738	0	0	67046146	0:0:00:00.00	765

To verify the size of the memory, use the **show chassis fpc**, **show chassis routing-engine**, and **show chassis ssb** commands. For more information about the output of these commands, see the *Junos OS System Basics and Services Command Reference*.

Table 57 on page 290 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of a T640 router.

Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	–	–	–	–	–	–	–	–
2	2	0	0	PEM 1	Running	29	–	–	–	–	–	–	–	–

Copyright © 2012, Juniper Networks, Inc.

Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router *(continued)*

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	1	1	0	Top left front fan	Running	0	–	–	–	–	–	–	–	–
4	1	2	0	Top left middle fan	Running	0	–	–	–	–	–	–	–	–
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	6	0	Top right rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom right rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	2	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	3	0	Middle blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	4	0	Top blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router *(continued)*

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	3	5	0	Second blower from top	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	Running	0	1	0	512	41	3	138367	0:18:56:48.81	512
7	2	1	0	FPC @ 1/0/* top temperature sensor	Running	35	0	0	0	0	0	0	0:18:56:48.81	0
7	2	2	0	FPC @ 1/1/* bottom temperature sensor	Running	32	0	0	0	0	0	0	0:18:56:48.81	0
7	6	0	0	FPC @ 5/*/*	Running	0	3	0	256	41	14	136976	0:18:57:02.71	256
7	6	1	0	FPC @ 5/0/* top temperature sensor	Running	44	0	0	0	0	0	0	0:18:57:02.71	0
7	6	2	0	FPC @ 5/1/* bottom temperature sensor	Running	33	0	0	0	0	0	0	0:18:57:02.71	0
7	8	0	0	FPC @ 7/*/*	Running	0	2	0	256	41	7	137963	0:18:56:52.85	256
7	8	1	0	FPC @ 7/0/* top temperature sensor	Running	38	0	0	0	0	0	0	0:18:56:52.85	0
7	8	2	0	FPC @ 7/1/* bottom temperature sensor	Running	33	0	0	0	0	0	0	0:18:56:52.85	0
9	1	0	0	Host 0	Running	35	0	0	2048	0	0	6963005	0:19:20:30.07	2048
9	2	0	0	Host 1	Standby	32	2	0	2048	0	0	24401100	2:19:46:51.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 57: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router *(continued)*

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
11	1	0	0	SCG 0	Running	36	0	0	0	0	0	0	0:0:00:00.00	0
11	2	0	0	SCG 1	Standby	35	0	0	0	0	0	0	0:0:00:00.00	0
12	1	0	0	CB 0	Running	36	0	0	0	0	0	0	0:0:00:00.00	0
12	2	0	0	CB 1	Standby	39	0	0	0	0	0	0	0:0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	1	0	128	40	0	142576	0:18:56:06.72	128
14	2	0	0	SPMB 1	Standby	39	0	0	128	40	0	142447	0:18:56:08.01	128
15	1	0	0	SIB 0	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0
15	2	0	0	SIB 1	Unknown	39	0	0	0	0	0	0	0:0:00:00.00	0
15	3	0	0	SIB 2	Unknown	39	0	0	0	0	0	0	0:0:00:00.00	0
15	4	0	0	SIB 3	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0
15	5	0	0	SIB 4	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0

Table 58 on page 293 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of a T320 router.

Table 58: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
2	1	0	0	PEM 0	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	1	0	Top left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	2	0	Top left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 58: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router *(continued)*

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Top right rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	1	0	Bottom left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	2	0	Bottom left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Bottom right rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	1	0	Rear tray top fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	2	0	Rear tray second fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	3	0	Rear tray middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	4	0	Rear tray fourth fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	5	0	Rear tray bottom fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	Running	0	1	0	256	41	7	6568428	(26190949) 3 days, 0:45:09.49	256
7	4	1	0	FPC @ 3/0/* top temperature sensor	Running	41	0	0	0	0	0	0	(26190949) 3 days, 0:45:09.49	0

Copyright © 2012, Juniper Networks, Inc.

Table 58: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router *(continued)*

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	4	2	0	FPC @ 3/1/* bottom temperature sensor	Running	37	0	0	0	0	0	0	(26190949) 3 days, 0:45:09.49	0
9	1	0	0	Host 0	Running	34	1	0	2048	0	0	32763001	(32763004) 3 days, 19:00:30.04	2048
9	2	0	0	Host 1	Standby	32	1	0	2048	0	0	110271900	(110271900) 12 days, 18:18:39.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	(0) 0:00:00.00	0
11	1	0	0	SCG 0	Running	33	0	0	0	0	0	0	(0) 0:00:00.00	0
11	2	0	0	SCG 1	Standby	31	0	0	0	0	0	0	(0) 0:00:00.00	0
12	1	0	0	CB 0	Running	37	0	0	0	0	0	0	(0) 0:00:00.00	0
12	2	0	0	CB 1	Standby	34	0	0	0	0	0	0	(0) 0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	0	0	128	40	0	6572381	(26186997) 3 days, 0:44:29.97	128
14	2	0	0	SPMB 1	Standby	36	1	0	128	40	0	6572465	(26186913) 3 days, 0:44:29.13	128
15	1	0	0	SIB 0	Standby	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	2	0	0	SIB 1	Running	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	3	0	0	SIB 2	Running	38	0	0	0	0	0	0	(0) 0:00:00.00	0

Table 59 on page 296 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of a T4000 router.

Table 59: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	2494800	0	0
2	1	0	0	PEM 1	Running	29	0	0	0	0	0	2494800	0	0
4	1	0	0	Front Top fan tray	Running	0	0	0	0	0	0	2494800	0	0
4	1	1	0	Top left front fan	Running	0	0	0	0	0	0	2494800	0	0
4	1	2	0	Top left middle fan	Running	0	0	0	0	0	0	2494800	0:0:00:00.00	0
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	1	6	0	Top right middle fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	2	0	0	Front Bottom Fan Tray	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	2	1	0	Bottom Left Front fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	2	0	0	Bottom left middle fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	1398451	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	1398452	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	1398452	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 59: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router *(continued)*

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	2	6	0	Bottom Right Rear fan	Running	0	0	0	0	0	0	1398452	0:0:00:00.00	0
4	3	0	0	Rear Fan Tray	Running	0	0	0	0	0	0	1398452	0:0:00:00.00	0
4	3	1	0	Rear Tray Top fan	Running	0	0	0	0	0	0	1398452	0:0:00:00.00	0
4	3	2	0	Rear Tray Second fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	3	0	Rear Tray Third fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	4	0	Rear Tray Fourth fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	6	0	Rear Tray Fifth fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	7	0	Rear Tray Sixth fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	8	0	Rear Tray Seventh fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
7	2	0	0	FPC Type 4-ES @ 1/*/*	Running	0	1	0	512	41	3	138367	0:18:56:48.81	512
7	2	1	0	FPC Type 4-ES @ 1/*/* top temperature sensor	Running	35	0	0	0	0	0	0	0:18:56:48.81	0
7	2	2	0	FPC Type 4-ES @ 1/*/* bottom temperature sensor	Running	32	0	0	0	0	0	0	0:18:56:48.81	0
7	3	0	0	FPC Type 5-3D @ 2/*/*	Running	0	3	0	256	41	14	136976	0:18:57:02.71	256

Copyright © 2012, Juniper Networks, Inc.

Table 59: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router *(continued)*

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	3	1	0	FPC Type 5-3D @ 2/*/* intake	Running	44	0	0	0	0	0	0	0:18:57:02.71	0
7	3	2	0	FPC Type 5-3D @ 2/*/* exhaust	Running	33	0	0	0	0	0	0	0:18:57:02.71	0
7	5	0	0	FPC Type 5-3D @ 4/*/*	Running	0	3	0	256	41	14	136976	0:18:57:02.71	256
7	6	1	0	FPC Type 5-3D @ 4/*/* intake	Running	44	0	0	0	0	0	0	0:18:57:02.71	0
7	5	2	0	FPC Type 5-3D @ 4/*/* exhaust	Running	33	0	0	0	0	0	0	0:18:57:02.71	0
8	3	1	0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/0/*	Running	0	0	0	0	0	0	2409241	0:18:56:52.85	0
8	3	2	0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/1/*	Running	33	0	0	0	0	0	0	0:18:56:52.85	0
8	5	1	0	PIC: 24x10GE (LAN/WAN) SFPP @ 4/0/*	Running	3	0	0	49	0	29624	0	0:18:56:52.85	0
8	5	2	0	PIC: 12x10GE (LAN/WAN) SFPP @ 4/1/*	Running	33	0	0	0	0	29240	0	0:18:56:52.85	0
9	1	0	0	Routing Engine 0	Running	35	3	0	8568	20	0	2494800	0:19:20:30.07	8950

Copyright © 2012, Juniper Networks, Inc.

Table 59: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router *(continued)*

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
9	2	0	0	Routing Engine 1	Standby	32	0	0	[illegible]	0	0	2402100	2:19:46:51.00	8
10	1	1	0	FPM GBUS	Running	30	0	0	0	0	0	1398474	0:0:00:00.00	0
10	1	2	0	FPM Display	Running	26	0	0	0	0	20	1398474	0:0:00:00.00	0
11	1	0	0	SCG 0	Running	36	0	0	0	0	0	577202	0:0:00:00.00	0
11	2	0	0	SCG 1	Standby	35	0	0	0	0	0	0	0:0:00:00.00	0
12	1	0	0	CB 0	Running	36	0	0	0	0	0	1398518	0:0:00:00.00	0
12	2	0	0	CB 1	Standby	39	0	0	0	0	0	0	0:0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	1	0	128	40	0	142576	0:18:56:06.72	128
14	1	0	0	SPMB 1	Standby	39	0	0	128	22	0	1396458	0:18:56:08.01	128
15	1	0	0	SIB 0	Unknown	40	0	0	0	22	0	1396458	0:0:00:00.00	0
15	2	0	0	SIB 1	Unknown	39	0	0	0	0	0	1389454	0:0:00:00.00	0
15	3	0	0	SIB 2	Unknown	39	0	0	0	0	0	386470	0:0:00:00.00	0
15	4	0	0	SIB 3	Unknown	40	0	0	0	0	0	1383482	0:0:00:00.00	0
15	5	0	0	SIB 4	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0

jnxRedundancyTable

The object identifier for the **jnxRedundancyTable** is **{jnxBoxAnatomy 14}**. This object shows the internal configuration settings for the redundant subsystems or components in the chassis.

Entries within the **jnxRedundancyTable** are represented by the **jnxRedundancyEntry** object, whose object identifier is **{jnxRedundancyEntry 1}**. This **jnxRedundancyEntry** contains the following objects, which describe the internal configuration settings for the redundant subsystems or components in the chassis:

- **jnxRedundancyContentsIndex**—The index value of an entry in **jnxRedundancyEntry**, whose object identifier is **{jnxContainersEntry 1}**.
- **jnxRedundancyL1Index**—The level-one index associated with the redundant component, whose object identifier is **{jnxContainersEntry 2}**.
- **jnxRedundancyL2Index**—The level-two index associated with the redundant component, whose object identifier is **{jnxContainersEntry 3}**.
- **jnxRedundancyL3Index**—The level-three index associated with the redundant component, whose object identifier is **{jnxContainersEntry 4}**.
- **jnxRedundancyDescr**—The description of the redundant component, whose object identifier is **{jnxContainersEntry 5}**.
- **jnxRedundancyConfig**—The election priority of redundancy configuration, whose object identifier is **{jnxContainersEntry 6}**.
- **jnxRedundancyState**—The current running state of the redundant component, whose object identifier is **{jnxContainersEntry 7}**.
- **jnxRedundancySwitchoverCount**—The total number of switchovers, defined as a change in the **jnxRedundancyState** from master to backup or vice versa, as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is **{jnxContainersEntry 8}**.
- **jnxRedundancySwitchoverTime**—The value of **sysUpTime** when the **jnxRedundancyState** was last switched over from master to backup or vice versa, whose object identifier is **{jnxContainersEntry 9}**.
- **jnxRedundancySwitchoverReason**—The reason for the last switchover to the redundant component, whose object identifier is **{jnxContainersEntry 10}**.
- **jnxKeepaliveHeartbeat**—The period of sending keepalive messages between the master and the backup subsystem, which is a system-wide preset value in seconds used by internal mastership resolution, whose object identifier is **{jnxContainersEntry 11}**.
- **jnxRedundancyKeepaliveTimeout**—The timeout period in seconds used by the watchdog timer before it initiates a switchover to the backup subsystem, whose object identifier is **{jnxContainersEntry 12}**.
- **jnxRedundancyKeepaliveElapsed**—The elapsed time since the redundant component received the last keepalive message from the outer subsystems, whose object identifier is **{jnxContainersEntry 13}**.
- **jnxRedundancyKeepaliveLoss**—The total number of keepalive messages lost between the master and the backup subsystems as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is **{jnxContainersEntry 14}**.

Table 60 on page 301 through Table 62 on page 304 provide examples of **jnxRedundancyEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxOperatingTable** objects:

 Copyright © 2012, Juniper Networks, Inc.

- Contents index—**jnxRedundancyContentsIndex**
- L1—**jnxRedundancyL1Index**
- L2—**jnxRedundancyL2Index**
- L3—**jnxRedundancyL3Index**
- Description—**jnxRedundancyDescr**
- Config—**jnxRedundancyConfig**
- State—**jnxRedundancyState**
- Count—**jnxRedundancySwitchoverCount**
- Time—**jnxRedundancySwitchoverTime**
- Reason—**jnxRedundancySwitchoverReason**
- Heartbeat—**jnxKeepaliveHeartbeat**
- Timeout—**jnxRedundancyKeepaliveTimeout**
- Elapsed—**jnxRedundancyKeepaliveElapsed**
- Loss—**jnxRedundancyKeepaliveLoss**

Table 60 on page 301 provides an example of **jnxRedundancyEntry** objects in the **jnxRedundancyTable** of an M20 router.

Table 60: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	Elapsed	Loss
6	1	0	0	SSB 0 Internet Processor II	Master	Master	0	3383	Never switched	0	0	0	0
6	2	0	0	SSB 1	Disabled	Disabled	0	0	Never switched	0	0	0	0
9	1	0	0	Routing Engine 0	Master	Master	1	421	User switched	3	300	1	0
9	2	0	0	Routing Engine 1	Backup	Backup	0	0	Other	0	0	0	0

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M20 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
 Slot 0:
  Current state      Master
  Election priority      Master (default)
  Temperature      26 degrees C / 78 degrees F
```

```
  DRAM                  768 Mbytes
  CPU utilization:
   User                  2 percent
   Background      0 percent
   Kernel            0 percent
   Interrupt         0 percent
   Idle                98 percent
  Model              teknor
  Serial ID      32000004f8ff1201
  Start time      2002-01-29 12:30:42 PST
  Uptime             21 hours, 17 minutes, 14 seconds
  Load averages:      1 minute  5 minute 15 minute
    0.03    0.02    0.00
Routing Engine status:
 Slot 1:
  Current state      Backup
  Election priority  Backup (default)
  DRAM                805306368 Mbytes
  CPU utilization:
   User                0 percent
   Background      0 percent
   Kernel            1 percent
   Interrupt         0 percent
   Idle                99 percent
  Model              teknor
  Serial ID      100000078c10df01
  Start time      2002-01-24 16:47:39 PST
  Uptime             5 days, 17 hours, 14 seconds
```

To verify SSB status, use the **show chassis ssb** command. Sample command output from an M20 router is as follows:

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
 State                    Master
 Temperature                24 degrees C / 75 degrees F
 CPU utilization              2 percent
 Interrupt utilization           0 percent
 Heap utilization             16 percent
 Buffer utilization           43 percent
 Total CPU DRAM              64 Mbytes
 Internet Processor II            Version 1, Foundry IBM, Part number 9
 Start time:                 2002-01-29 12:32:24 PST
 Uptime:                   21 hours, 30 minutes, 53 seconds
Slot 1 information:
 State                    Backup
```

Table 61 on page 302 provides an example of **jnxRedundancyEntry** objects in the **jnxRedundancyTable** of a T640 router.

Table 61: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Router

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	3	0:18:55:49.42	User switched	20	300	1	0

Copyright © 2012, Juniper Networks, Inc.

Table 61: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Router *(continued)*

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	Elapsed	Loss
9	2	0	0	Host 1	Backup	Backup	0	0:0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Unknown	Backup	1	0:0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	4	0	0	SIB 3	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	5	0	0	SIB 4	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state        Master
    Election priority       Master (default)
    Temperature       35 degrees C / 95 degrees F
    DRAM            2048 MB
    CPU utilization:
      User             1 percent
      Background       0 percent
      Kernel           5 percent
      Interrupt        0 percent
      Idle            94 percent
    Model               unknown
    Start time     2002-03-31 14:26:49 PST
    Uptime         19 hours, 22 minutes, 13 seconds
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority              Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
    Start time     2002-03-29 14:00:18 PST
    Uptime         2 days, 19 hours, 48 minutes, 32 seconds
```

Table 62 on page 304 provides an example of **jnxRedundancyEntry** objects in the **jnxRedundancyTable** of a T320 router.

Table 62: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Timeout	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	6	(26185188)3 days, 0:44:11.88	User switched	20	300	1	0
9	2	0	0	Host 1	Backup	Backup	0	(0) 0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Backup	Backup	1	(0) 0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from a T320 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state  Master
    Election priority              Master (default)
    Temperature 34 degrees C / 93 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     1 percent
      Interrupt  0 percent
      Idle      98 percent
    Model          RE-3.0
    Start time     2002-04-05 14:43:16 PST
    Uptime         17 days, 23 hours, 3 minutes, 47
seconds
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority              Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
```


Copyright © 2012, Juniper Networks, Inc.

```
    Start time      2002-03-27 15:25:07 PST
    Uptime          26 days, 22 hours, 21 minutes, 44 seconds
```

Table 63 on page 305 provides an example of **jnxRedundancyEntry** objects in the **jnxRedundancyTable** of a T4000 router.

Table 63: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T4000 Router

Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heart beat	Time out	Elapsed	Loss
9	1	0	0	Routing Engine 0	Master	Master	0	0:18:55:49.42	User switched	20	300	1	0
9	2	0	0	Routing Engine 1	Backup	Backup	0	0:0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Unknown	Backup	0	0:0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Unknown	Master	0	0:0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Unknown	Master	0	0:0:00:00.00	0	0	0	0	0
15	4	0	0	SIB 3	Unknown	Master	0	0:0:00:00.00	0	0	0	0	0
15	5	0	0	SIB 4	Unknown	Master	0	0:0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from a T4000 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 33 degrees C / 91 degrees F
    CPU temperature             50 degrees C / 122 degrees F
    DRAM                      8960 MB
    Memory utilization          18 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     4 percent
      Interrupt                  1 percent
      Idle                      95 percent
    Model                        RE-DUO-1800
    Serial ID                    P737F-002248
    Start time                   2012-02-09 22:49:53 PST
    Uptime                       2 hours, 21 minutes, 35 seconds
    Last reboot reason           Router rebooted after a normal shutdown.
    Load averages:               1 minute   5 minute  15 minute
                                     0.00       0.04       0.00
Routing Engine status:
  Slot 1:
    Current state                Backup
    Election priority            Backup (default)
```

```
Temperature                 32 degrees C / 89 degrees F
CPU temperature             46 degrees C / 114 degrees F
DRAM                      8960 MB
Memory utilization          24 percent
CPU utilization:
  User                       0 percent
  Background                 0 percent
  Kernel                     0 percent
  Interrupt                  0 percent
  Idle                      99 percent
Model                           RE-DUO-1800
Serial ID                       P737F-002653
Start time                      2012-02-08 20:12:51 PST
Uptime                          1 day, 4 hours, 58 minutes, 28 seconds
Last reboot reason              Router rebooted after a normal shutdown.
```

jnxFruTable

The object identifier for the **jnxFruTable** is **{jnxBoxAnatomy 15}**. This object shows the status of field-replaceable units (FRUs) in the chassis.

Entries within the **jnxFruTable** are represented by the **jnxFruEntry** object, whose object identifier is **{jnxFruEntry 1}**. This **jnxFruEntry** object contains the following objects, which describe the FRUs in the chassis:

- **jnxFruContentsIndex**—The index value of an entry in **jnxFruEntry**, whose object identifier is **{jnxFruEntry 1}**.
- **jnxFruL1Index**—The level-one index associated with the FRU, whose object identifier is **{jnxFruEntry 2}**.
- **jnxFruL2Index**—The level-two index associated with the FRU, whose object identifier is **{jnxFruEntry 3}**.
- **jnxFruL3Index**—The level-three index associated with the FRU, whose object identifier is **{jnxFruEntry 4}**.
- **jnxFruName**—The name or detailed description of the FRU, whose object identifier is **{jnxFruEntry 5}**.
- **jnxFruType**—The FRU type, whose object identifier is **{jnxFruEntry 6}**. The FRU type is represented by one of the following values:
  - **other(1)**
  - **clockGenerator(2)**
  - **flexiblePicConcentrator(3)**
  - **switchingAndForwardingModule(4)**
  - **controlBoard(5)**
  - **routingEngine(6)**
  - **powerEntryModule(7)**
  - **frontPanelModule(8)**

Copyright © 2012, Juniper Networks, Inc.

  - **switchInterfaceBoard(9)**
  - **processorMezzanineBoardForSIB(10)**
  - **portInterfaceCard(11)**
  - **craftInterfacePanel(12)**
  - **fan(13)**
  - **lineCardChassis(14)**
  - **forwardingEngineBoard(13)**
  - **protectedSystemDomain(13)**
- **jnxFruSlot**—The slot number of the FRU, whose object identifier is **{jnxFruEntry 7}**. This is equivalent to **jnxFruL1Index**. The slot number is zero if unavailable or inapplicable.
- **jnxFruState**—The current state of the FRU, whose object identifier is **{jnxFruEntry 8}**. The FRU state can be any of the following:
  - **unknown(1)**
  - **empty(2)**
  - **present(3)**
  - **ready(4)**
  - **announceOnline(5)**
  - **online(6)**
  - **announceOffline(7)**
  - **offline(8)**
  - **diagnostic(9)**
  - **standby(10)**
- **jnxFruTemp**—The temperature of the FRU, in degrees Celsius, whose object identifier is **{jnxFruEntry 9}**. The value is zero if unavailable or inapplicable.
- **jnxFruOfflineReason**—The reason the FRU is offline, whose object identifier is **{jnxFruEntry 10}**. The reason is represented by the following values:
  - **unknown(1)**—Unknown or other
  - **none(2)**—None
  - **error(3)**—Error
  - **noPower(4)**—No power
  - **configPowerOff(5)**—Configured to power off

  - **configHoldInReset(6)**—Configured to hold in reset
  - **cliCommand(7)**—Brought offline by a command-line interface (CLI) command
  - **buttonPress(8)**—Brought offline by button press
  - **cliRestart(9)**—Restarted by CLI command
  - **overtempShutdown(10)**—Overtemperature shutdown
  - **masterClockDown(11)**—Master clock down
  - **singleSfmModeChange(12)**—Single SFM mode change
  - **packetSchedulingModeChange(13)**—Packet scheduling mode change
  - **physicalRemoval(14)**—Physical removal
  - **unresponsiveRestart(15)**—Restarting unresponsive board
  - **sonetClockAbsent(16)**—SONET out clock absent

- **jnxFruLastPowerOff**—The value of **sysUpTime** when this subject was last powered off, whose object identifier is **{jnxFruEntry 11}**. The value is zero if unavailable or inapplicable.
- **jnxFruLastPowerOn**—The value of **sysUpTime** when this subject was last powered on, whose object identifier is **{jnxFruEntry 12}**. The value is zero if unavailable or inapplicable.
- **jnxFruPowerUpTime**—The time interval in 10-millisecond periods that this subject has been up and running since the last power-on time, whose object identifier is **{jnxFruEntry 13}**. The value is zero if unavailable or inapplicable.
- **jnxFruChassisId**—The chassis type of this subject. The object identifier for this object is **{jnxFruEntry 14}**.
- **jnxFruChassisDescr**—The textual description for the chassis type of this subject. The object identifier is **{jnxFruEntry 15}**.
- **jnxFruPsdAssignment**—The protected system domain (PSD) assignment for this subject. The object identifier is **{jnxFruEntry 16}**.

Table 64 on page 309 through Table 69 on page 333 provide examples of **jnxFruEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxFruEntry** objects:

- Contents index—**jnxFruContentsIndex**
- L1—**jnxFruL1Index**
- L2—**jnxFruL2Index**
- L3—**jnxFruL3Index**
- Name—**jnxFruName**
- Type—**jnxFruType**

 Copyright © 2012, Juniper Networks, Inc.

- Slot—**jnxFruSlot**
- State—**jnxFruState**
- Temp—**jnxFruTemp**
- Offline—**jnxFruOffline**
- PowerOff—**jnxFruPowerOff**
- PowerOn—**jnxFruPowerOn**
- Uptime—**jnxFruPowerUpTime**

Table 64 on page 309 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M10 router.

Table 64: jnxFruContents Objects in the jnxFruTable of an M10 Router

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntryModule	1	Online	0	None	0:0:00:00.00	0:0:11:08.73	264319
2	2	0	0	Power supply B	powerEntryModule	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	1	0	Left fan 1	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Left fan 2	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Left fan 3	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Left fan 4	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	FEB Internet Processor II	controlBoard	1	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x Monitor @ 0/1/*	portInterfaceCard	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 64: jnxFruContents Objects in the jnxFruTable of an M10 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	3	0	PIC: 1x OC-12 ATM, MM @ 0/2/*	portInterfaceCard	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 4x T3 @ 0/3/*	portInterfaceCard	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 4x OC-3 SONET, SMIR @ 1/0/*	portInterfaceCard	2	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 2x OC-3 ATM, MM @ 1/3/*	portInterfaceCard	2	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M10 router is as follows:.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                58974             M10
Midplane         REV 03   710-001950   HB1590
Power Supply A   Rev 03   740-002498   LK33505           DC
Display          REV 04   710-001995   HE8442
Routing Engine   REV 01   740-003239   9001025728        RE-2.0
FEB              REV 12   710-001948   HA4221            Internet Processor II
FPC 0
  PIC 1          REV 01   750-004188   AR2912            1x Monitor
  PIC 2          REV 04   750-001551   AN7869            1x OC-12 ATM, MM
  PIC 3          REV 02   750-002485   AN2803            4x T3
FPC 1
  PIC 0          REV 03   750-002970   HF2293            4x OC-3 SONET, SMIR
```


Copyright © 2012, Juniper Networks, Inc.

```
  PIC 1          REV 03   750-002971   HA8094            4x OC-3 SONET, MM
  PIC 2          REV 03   750-002977   HD9352            2x OC-3 ATM, MM
  PIC 3          REV 03   750-002977   HD9393            2x OC-3 ATM, MM
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from an M10 router is as follows:

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            24      3          1        64        44         17
  1  Online            24      3          1        64        44         17
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M10 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                 26 degrees C / 78 degrees F
    DRAM                       768 MB
    Memory utilization           9 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     0 percent
      Interrupt                  0 percent
      Idle                     100 percent
    Model                          RE-2.0
    Serial ID                      b7000007c81ce801
    Start time                     2002-06-21 09:33:45 PDT
    Uptime                         3 days, 1 hour, 23 minutes, 27 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.07       0.03       0.01
```

To verify FEB status, use the **show chassis feb** command. Sample command output from an M10 router is as follows:

```
user@host> show chassis feb
FEB status:
  Temperature                 24 degrees C / 75 degrees F
  CPU utilization              3 percent
  Interrupt utilization        1 percent
  Heap utilization            17 percent
  Buffer utilization          44 percent
  Total CPU DRAM              64 MB
  Internet Processor II        Version 1, Foundry IBM, Part number 9
  Start time:       2002-06-21 09:45:46 PDT
  Uptime:       3 days, 1 hour, 11 minutes, 33 seconds
```

Table 65 on page 311 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M20 router.

Table 65: JnxFruContents Objects in the jnxFruTable of an M20 Router

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	**powerEntryModule**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

Table 65: JnxFruContents Objects in the jnxFruTable of an M20 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	2	0	0	Power supply B	**powerEntryModule**	2	Online	25	None	0:0:00:00.00	0:0:00:43.45	24993357
4	1	0	0	Rear fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Front upper fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Front middle fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Front bottom fan	**fan**	4	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SSB 0	**controlBoard**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
6	2	0	0	SSB 1 Internet Processor I	**controlBoard**	2	Online	29	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	**flexible PicConcentrator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	**flexible PicConcentrator-**	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	**flexible PicConcentrator**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	**flexible PicConcentrator**	4	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	**portInterfaceCard**	1	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	**portInterfaceCard**	1	Offline	28	None	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	**portInterfaceCard**	1	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	**portInterfaceCard**	1	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x Tunnel @ 1/0/*	**portInterfaceCard**	2	Ready	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 65: JnxFruContents Objects in the jnxFruTable of an M20 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	2	0	PIC: 4x T3 @ 1/1/*	**portInterfaceCard**	2	Ready	0	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x G/E, 1000Base-SX @ 1/3/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	**portInterfaceCard**	3	Offline	27	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	**portInterfaceCard**	4	Ready	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	**portInterfaceCard**	4	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 2x OC-3 SONET, SMIR @ 3/2/*	**portInterfaceCard**	4	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	**portInterfaceCard**	4	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	**routingEngine**	1	Online	25	None	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	**routingEngine**	2	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
10	1	0	0	Front panel display	**frontPanelModule**	1	Online	0	None	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M20 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                20200             M20
Backplane        REV 07   710-001517   AB5911
Power Supply B   Rev 02   7            000240            AC
Display          REV 04   710-001519   AD1903
Routing Engine 0 REV01    740          umeshk            RE-2.0
Routing Engine 1                       270000078ba48501  RE-2.0
SSB slot 0       N/A      N/A          N/A               backup
SSB slot 1       REV 04   710-001411   AD0281            Internet Processor I
FPC 1            REV 01   710-001292   AC9230
  PIC 0          REV 01   750-001323   AA2812            1x Tunnel
  PIC 1          REV 01   750-002963   AK8586            4x T3
  PIC 2          REV 03   750-000612   AM8116            2x OC-3 ATM, MM
  PIC 3          REV 08   750-001072   AB9884            1x G/E, 1000 BASE-SX
FPC 3            REV 01   710-001197   AA8661
  PIC 2          REV 01   750-003748   HE9734            2x OC-3 SONET, SMIR
```

```
user@host> show chassis environment
Class Item                   Status     Measurement
Power Power Supply A         Absent
      Power Supply B         OK         25 degrees C / 77 degrees F
Temp  FPC 1                  OK         27 degrees C / 80 degrees F
      FPC 3                  OK         28 degrees C / 82 degrees F
      SSB 1                  OK         29 degrees C / 84 degrees F
      Backplane              OK         23 degrees C / 73 degrees F
      Routing Engine 0       OK         25 degrees C / 77 degrees F
      Routing Engine 1       OK         24 degrees C / 75 degrees F
Fans  Rear Fan               OK         Spinning at normal speed
      Front Upper Fan        OK         Spinning at normal speed
      Front Middle Fan       OK         Spinning at normal speed
      Front Bottom Fan       OK         Spinning at normal speed
Misc  Craft Interface        OK
```

```
user@host> show chassis fpc
                     Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty              0      0          0         0         0          0
  1  Online            27      8          7         8         9         14
  2  Empty              0      0          0         0         0          0
  3  Online            28      0          0         8         8         14
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M10 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 25 degrees C / 77 degrees F
    DRAM                       768 MB
    Memory utilization           8 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     1 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                          RE-2.0
    Serial ID                      ba0000061779d601
```


Copyright © 2012, Juniper Networks, Inc.

```
    Start time                      2002-06-21 15:37:36 PDT
    Uptime                          2 days, 21 hours, 27 minutes, 25 seconds
    Load averages:                  1 minute   5 minute  15 minute
                                        0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state                   Backup
    Election priority               Backup (default)
    Temperature                  24 degrees C / 75 degrees F
    DRAM                        768 MB
    Memory utilization            9 percent
    CPU utilization:
      User                        0 percent
      Background                  0 percent
      Kernel                      0 percent
      Interrupt                   0 percent
      Idle                       99 percent
    Model                         RE-2.0
    Serial ID                     270000078ba48501
    Start time                    2002-06-17 14:30:21 PDT
    Uptime                        6 days, 22 hours, 34 minutes, 28 seconds
```

To verify SSB status, use the **show chassis ssb** command. Sample command output from an M10 router is as follows:

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
  State                           Backup
Slot 1 information:
  State                           Master
  Temperature                  29 degrees C / 84 degrees F
  CPU utilization               1 percent
  Interrupt utilization         0 percent
  Heap utilization              8 percent
  Buffer utilization           43 percent
  Total CPU DRAM               64 MB
  Internet Processor I            Version 1, Foundry IBM, Part number 3
  Start time:                     2002-06-21 15:38:53 PDT
  Uptime:                         2 days, 21 hours, 26 minutes, 26 seconds
```

Table 66 on page 315 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M160 router.

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	**powerEntryModule**	1	Online	0	None	0:00:00.00	0:00:12.83	6906955
2	2	0	0	PEM 1	**powerEntryModule**	2	Online	0	None	0:00:00.00	0:00:12.83	6906955
4	1	0	0	Front top blower	**fan**	1	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	1	0	Fan tray front left	**fan**	2	Present	0	None	0:00:00.00	0:00:00.00	0

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	2	2	0	Fan tray front right	**fan**	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	3	0	Fan tray rear left	**fan**	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	4	0	Fan tray rear right	**fan**	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	3	0	0	Rear top blower	**fan**	3	Present	0	None	0:00:00.00	0:00:00.00	0
4	4	0	0	Rear bottom blower	**fan**	4	Present	0	None	0:00:00.00	0:00:00.00	0
6	1	1	0	SFM 0 SPP	**switchingAnd ForwardingMode**	1	Online	35	None	0:00:03.13	0:00:00.00	0
6	1	2	0	SFM 0 SPR Internet Processor II	**switchingAnd ForwardingMode**	1	Online	35	None	0:00:03.13	0:00:00.00	0
6	2	1	0	SFM 1 SPP	**switchingAnd ForwardingMode**	2	Empty	0	None	0:00:00.00	0:00:00.00	0
6	2	2	0	SFM 1 SPR	**switchingAnd ForwardingMode**	2	Empty	0	None	0:00:00.00	0:00:00.00	0
6	3	1	0	SFM 2 SPP	**switchingAnd ForwardingMode**	3	Online	44	None	0:00:03.20	0:00:00.00	0
6	3	2	0	SFM 2 SPR Internet Processor II	**switchingAnd ForwardingMode**	3	Online	44	None	0:00:03.20	0:00:00.00	0
6	4	1	0	SFM 3 SPP	**switchingAnd ForwardingMode**	4	Offline	0	Config-ured to power off	0:00:03.22	0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
6	4	2	0	SFM 3 SPR	**switchingAnd ForwardingMode**	4	Offline	0	Config-ured to power off	0:00:03.22	0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	**flexiblePic Concentrator**	1	Offline	0	Config-ured to power off	0:00:02.28	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	**flexiblePic Concentrator**	2	Offline	0	Error	0:13:08.12	0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	**flexiblePic Concentrator**	3	Online	30	None	0:00:02.32	0:00:00.00	0
7	4	0	0	FPC: 1x OC-192 SM LR @ 3/*/*	**flexiblePic Concentrator**	4	Online	41	None	0:00:02.34	0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	**flexiblePic Concentrator**	5	Empty	0	None	0:00:00.00	0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	**flexiblePic Concentrator**	6	Offline	0	Config ured to power off	0:00:02.37	0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	**flexiblePic Concentrator**	7	Empty	0	None	0:00:00.00	0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	**flexiblePic Concentrator**	8	Online	41	None	0:00:03.11	0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	**portInterfaceCard**	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	**portInterface Card**	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	**portInterfaceCard**	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	**portInterfaceCard**	1	Online	40	None	0:00:00.00	0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	1	0	PIC: @ 1/0/*	**portInterfaceCard**	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	**portInterfaceCard**	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	**portInterfaceCard**	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	**portInterfaceCard**	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	**portInterfaceCard**	3	Offline	0	Config-ured to power off	0:00:02.28	0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	**portInterfaceCard**	3	Offline	0	Config-ured to power off	0:00:02.28	0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	**portInterfaceCard**	3	Offline	0	Config ured to power off	0:00:02.28	0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	**portInterfaceCard**	3	Offline	0	Config-ured to power off	0:00:02.28	0:00:00.00	0
8	4	1	0	PIC: 1x OC-192 SM LR @ 3/0/*	**portInterfaceCard**	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	2	0	PIC continued	**portInterfaceCard**	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	3	0	PIC continued	**portInterfaceCard**	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	4	0	PIC continued	**portInterfaceCard**	4	Offline	0	Error	0:13:08.12	0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	1	0	PIC: @ 4/0/*	**portInterfaceCard**	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	**portInterfaceCard**	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	**portInterfaceCard**	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	**portInterfaceCard**	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	**portInterfaceCard**	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	**portInterfaceCard**	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	**portInterfaceCard**	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	**portInterfaceCard**	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	**portInterfaceCard**	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	**portInterfaceCard**	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	**portInterfaceCard (11)**	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	**portInterfaceCard (11)**	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	8	1	0	PIC: 1x OC-12 SONET, SMIR @ 7/0/*	**portInterfaceCard**	8	Offline	0	Configured to power off	0:00:02.37	0:00:00.00	0
8	8	2	0	PIC: 4x E3 @ 7/1/*	**portInterfaceCard**	8	Offline	0	Configured to power off	0:00:02.37	0:00:00.00	0

Table 66: jnxFruContents Objects in the jnxFruTable of an M160 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	3	0	PIC: 1x OC-12 SONET, MM @ 7/2/* jnxFruName	**portInterfaceCard**	8	Offline	0	Configured to power off	0:00:02.37	0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	**portInterfaceCard**	8	Offline	0	Configured to power off	0:00:02.37	0:00:00.00	0
9	1	0	0	Routing Engine 0	**routingEngine**	1	Online	31	None	0:00:00.00	0:00:00.00	0
9	2	0	0	Routing Engine 1	**routingEngine**	2	Present	0	None	0:00:00.00	0:00:00.00	0
10	1	1	0	FPM CMB	**frontPanelModule**	1	Online	28	None	0:00:00.00	0:00:00.00	0
10	1	2	0	FPM Display	**frontPanelModule**	1	Online	28	None	0:00:00.00	0:00:00.00	0
11	1	0	0	PCG 0	**clockGenerator**	1	Online	40	None	0:00:00.00	0:00:00.00	0
11	2	0	0	PCG 1	**clockGenerator**	2	Online	46	None	0:00:00.00	0:00:00.00	0
12	1	0	0	MCS 0	**controlBoard**	1	Online	47	None	0:00:00.00	0:00:00.00	0
12	2	0	0	MCS 1	**controlBoard**	2	Empty	0	None	0:00:00.00	0:00:00.00	0
13	1	0	0	CIP	**craftInterfacePanel-**	1	Present	0	None	0:00:00.00	0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M160 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                47                M160
Midplane         REV 02   710-001245   AB4113
FPM CMB          REV 01   710-001642   AA9721
FPM Display      REV 01   710-001647   AA2995
CIP              REV 02   710-001593   AA9886
PEM 0            Rev 01   740-001243   KJ35782           DC
PEM 1            Rev 01   740-001243   kj35756           DC
PCG 0            REV 01   710-001568   AA9796
```


Copyright © 2012, Juniper Networks, Inc.

```
PCG 1            REV 01   710-001568   AA9895
Routing Engine 0 REV01    740-003239   AARCHOO           RE-2.0
Routing Engine 1
MCS 0            REV 03   710-001226   AA9779
SFM 0 SPP        REV 07   710-001228   AE5504
SFM 0 SPR        REV 03   710-002189   AE4707            Internet Processor II
SFM 2 SPP        REV 06   710-001228   AB3133
SFM 2 SPR        REV 01   710-002189   AB2941            Internet Processor II
SFM 3 SPP        REV 07   710-001228   AV3167
SFM 3 SPR        REV 04   710-002189   AV3439            Internet Processor II
FPC 0            REV 02   710-001611   AA9518            FPC Type 2
  CPU            REV 02   710-001217   AA9572
FPC 1            REV 03   710-001255   AA9812            FPC Type 1
  CPU
FPC 2            REV 02   710-001611   AA9527            FPC Type 2
  CPU            REV 02   710-001217   AA9592
FPC 3            REV 01   710-003061   HB2029            FPC Type OC192
  CPU            REV 05   710-001217   AF5950
  PIC 0          REV 01   750-003063   HB2029            1x OC-192 SM LR
FPC 5            REV 01   710-001255   AA2914            FPC Type 1
  CPU            REV 02   710-001217   AA2893
FPC 7            REV 03   710-001255   AA9809            FPC Type 1
  CPU            REV 02   710-001217   AA9573
  PIC 0          REV 04   750-000613   AA0374            1x OC-12 SONET, SMIR
  PIC 1          REV 02   750-E3-PIC   AC1903            4x E3
  PIC 2          REV 02   750-001020   AA8944            1x OC-12 SONET, MM
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from an M160 router is as follows:

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Announce offline   0      0          0         0         0          0
  1  Present            0      0          0         0         0          0
  2  Online            32      4          0        32         1         39
  3  Online            44      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Offline         --- Chassis connection dropped ---
  6  Empty              0      0          0         0         0          0
  7  Online            42      4          0        32         1         40
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M160 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                       768 MB
    Memory utilization          10 percent
    CPU utilization:
      User                       1 percent
      Background                 0 percent
      Kernel                    10 percent
      Interrupt                  3 percent
      Idle                      87 percent
    Model                          RE-2.0
    Serial ID                      0c000004f8d26401
```

```
    Start time                      2002-06-14 14:39:03 PDT
    Uptime                          11 minutes, 46 seconds
    Load averages:                  1 minute   5 minute  15 minute
                                        0.18       0.19       0.14
Routing Engine status:
  Slot 1:
    Current state                   Present
```

To verify SFM status, use the **show chassis sfm** command. Sample command output from an M160 router is as follows:

```
user@host> show chassis sfm
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            35      1          0         64        16         46
  1  Empty              0      0          0          0         0          0
  2  Online            47      1          0         64        16         45
  3  Online            50      1          0         64        16         45
Packet scheduling mode : Disabled
```

Table 67 on page 322 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M40 router.

Table 67: jnxFruContents Objects in the jnxFruTable of an M40 Router

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	**powerEntry Module**	1	Online	0	None	0:0:00:00.00	0:0:00:00.00	101974
2	2	0	0	Power supply B	**powerEntry Module**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
3	1	0	0	Top impeller	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
3	2	0	0	Bottom impeller	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	0	0	Rear left fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Rear center fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear right fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
5	1	0	0	Host controller	**routingEngine**	1	Online	37	None	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SCB Internet Processor I	**controlBoard**	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 67: jnxFruContents Objects in the jnxFruTable of an M40 Router *(continued)*

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	1	0	0	FPC @ 0/*/*	**flexiblePic Concentrator**	1	Online	28	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	**flexible PicConcentrator**	2	Online	29	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	**flexible PicConcentrator**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	**flexiblePic Concentrator**	4	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	**flexiblePic Concentrator**	5	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	**flexiblePic Concentrator**	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	7	0	0	FPC: 1x OC-48 SONET, SMIR @ 6/*/*	**flexiblePic Concentrator**	7	Online	28	None	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	**flexible PicConcentrator**	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: 1x G/E, 1000Base-SX @ 0/0/*	**portInterfaceCard**	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x Tunnel @ 0/1/*	**portInterfaceCard**	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x T1, RJ48 @ 0/2/*	**portInterfaceCard**	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x COC12, SMIR @ 0/3/*	**portIntefaceCard**	1	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 2x OC-3 ATM, MM @ 1/0/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 67: jnxFruContents Objects in the jnxFruTable of an M40 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x T3 @ 1/2/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x CSTM1, SMIR @ 1/3/*	**portInterfaceCard**	2	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	**portInterfaceCard**	3	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	**portInterfaceCard**	4	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: 4x F/E, 100 BASE-TX @ 3/1/*	**portInterfaceCard**	4	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 1x 800M Crypto @ 3/2/*	**portInterfaceCard**	4	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: 1x CT3-NxDS0 @ 3/3/*	**portInterfaceCard**	4	Ready	24	None	0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	**portInterfaceCard**	5	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	**portInterfaceCard**	5	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 67: jnxFruContents Objects in the jnxFruTable of an M40 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	3	0	PIC: @ 4/2/*	**portInterfaceCard**	5	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	**flexible PicConcentrator**	5	Ready	27	None	0:0:00:00.00	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	**portInterfaceCard**	6	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	**portInterfaceCard**	6	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	**portInterfaceCard**	6	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	**portInterfaceCard**	6	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: 1x OC-48 SONET, SMIR @ 6/0/*	**portInterfaceCard**	7	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC continued	**portInterfaceCard**	7	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC continued	**portInterfaceCard**	7	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC continued	**portInterfaceCard**	7	Ready	28	None	0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	**portInterfaceCard**	8	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	**portInterfaceCard**	8	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	**portInterfaceCard**	8	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	**portInterfaceCard**	8	Offline	0	None	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	**routingEngine**	1	Online	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M40 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis
Backplane        REV 03   710-000073   AA2005
Power Supply A   Rev A    740-000235   000119            DC
Maxicab          REV 04   710-000229   AA0691
Minicab          REV 02   710-000482   AA0270
Display          REV 06   710-000150   AA1042
Routing Engine                                           RE-1.0
SCB              REV 07   710-000075   AA1033            Internet Processor I
FPC 0            REV 01   710-001292   AB8159
  PIC 0          REV 08   750-001072   AP5525            1x G/E, 1000 BASE-SX
  PIC 1          REV 01   750-001323   AB1645            1x Tunnel
  PIC 2          REV 01   750-002953   AD9083            4x T1, RJ48
  PIC 3          REV 03   750-001190   AE2907            1x COC12, SMIR
FPC 1            REV 10   710-000175   AA7219
  PIC 0          REV 03   750-002977   HD9331            2x OC-3 ATM, MM
  PIC 1          REV 04   750-002971   HC8020            4x OC-3 SONET, MM
  PIC 2          REV 02.1 710-000608   AA1592            2x T3
  PIC 3          REV 05   750-003248   AD9648            1x CSTM1, SMIR
FPC 3            REV 10   710-000175   AA4782
  PIC 1          REV 04   750-002992   HC3974            4x F/E, 100 BASE-TX
  PIC 2          REV 03   750-003844   AY4806            1x 800M Crypto
  PIC 3          REV 03   750-004743   BD9433            1x CT3-NxDS0
FPC 4            REV 01   710-001292   AC5265
FPC 6            REV 01   710-001292   AB7485
  PIC 0          REV 03   750-000617   AA4566            1x OC-48 SONET, SMIR
```

```
user@host> show chassis environment
Class Item                  Status     Measurement
Power Power Supply A        OK
      Power Supply B        Absent
Temp  FPC 0                 OK         28 degrees C / 82 degrees F
      FPC 1                 OK         29 degrees C / 84 degrees F
      FPC 3                 OK         24 degrees C / 75 degrees F
      FPC 4                 OK         27 degrees C / 80 degrees F
      FPC 6                 OK         28 degrees C / 82 degrees F
      SCB                   OK         27 degrees C / 80 degrees F
      Backplane @ A1        OK         30 degrees C / 86 degrees F
      Backplane @ A2        OK         26 degrees C / 78 degrees F
      Routing Engine        OK         37 degrees C / 98 degrees F
Fans  Top Impeller          OK         Spinning at normal speed
      Bottom impeller       OK         Spinning at normal speed
      Rear Left Fan         OK         Spinning at normal speed
      Rear Center Fan       OK         Spinning at normal speed
      Rear Right Fan        OK         Spinning at normal speed
Misc  Craft Interface       OK
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from an M40 router is as follows:

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            28      2          0         8        11         14
  1  Online            29      7          0         8        21         14
  2  Empty              0      0          0         0         0          0
```


Copyright © 2012, Juniper Networks, Inc.

```
  3  Online          24     17          0          8         22         15
  4  Online          27      1          0          8          6         13
  5  Empty            0      0          0          0          0          0
  6  Online          28      1          0          8          7         15
  7  Empty            0      0          0          0          0          0
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M40 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                 37 degrees C / 98 degrees F
    DRAM                              256 MB
    Memory utilization          19 percent
    CPU utilization:
      User                       1 percent
      Background                 0 percent
      Kernel                     3 percent
      Interrupt                  1 percent
      Idle                         96 percent
    Model         RE-1.0
    Start time                   2002-06-24 17:28:30 UTC
    Uptime                       20 minutes, 30 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.00       0.04       0.11
```

To verify SCB status, use the **show chassis scb** command. Sample command output from an M40 router is as follows:

```
user@host> show chassis scb
SCB status:
  Temperature                         27 degrees C / 80 degrees F
  CPU utilization                     3 percent
  Interrupt utilization                 0 percent
  Heap utilization          9 percent
  Buffer utilization          44 percent
  Total CPU DRAM                        64 MB
  Internet Processor I           Version 1, Foundry IBM, Part number 3
  Start time:                    2002-06-24 17:30:10 UTC
  Uptime:                        19 minutes, 8 seconds
```

Table 68 on page 327 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M40e router.

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	**powerEntryModule**	1	Present	0	None	0:0:00:00.00	0:0:00:25.99	208927
2	2		0	PEM 1	**powerEntryModule**	2	Online	0	None	0:0:00:00.00	0:0:00:25.99	208928
4	1	0	0	Front top blower	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Fan tray front left	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	2	2	0	Fan tray front right	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Fan tray rear left	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Fan tray rear right	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear top blower	**fan**	3	Present		None	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Rear bottom blower	**fan**	4	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
6	1	1	0	SFM 0 SPP	**switchingAnd ForwardingModule**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
6	1	2	0	SFM 0 SPR	**switchingAnd ForwardingModule**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
6	2	1	0	SFM 1 SPP	**switchingAnd ForwardingModule**	2	Online	42	None	0:0:00:21.95	0:0:00:00.00	0
6	2	2	0	SFM 1 SPR Internet Processor II	**switchingAnd ForwardingModule**	2	Online	42	None	0:0:00:21.95	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	**flexiblePic Concentrator**	1	Online	41	None	0:0:00:21.85	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	**flexible PicConcentrator**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	**flexible PicConcentrator**	3	Online	43	None	0:0:00:21.87	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	**flexible PicConcentrator**	4	Online	38	None	0:0:00:21.89	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	**flexiblePic Concentrator**	5	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	**flexiblePic Concentrator**	6	Online	46	None	0:0:00:21.91	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	7	0	0	FPC @ 6/*/*	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	**flexiblePic Concentrator**	8	Online	44	None	0:0:00:21.93	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	**portInterfaceCard**	1	Online	45	None	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x OC-12 SONET, MM @ 0/1/*	**portInterfaceCard**	1	Online	45	None	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x CT3 @ 0/2/*	**portInterfaceCard**	1	Online	45	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x Multi Link(32) @ 0/3/*	**portInterfaceCard**	1	Online	45	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	**portInterfaceCard**	2	Online	50	None	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	**portInterfaceCard**	2	Online	50	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	**portInterfaceCard**	2	Online	50	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	**portInterfaceCard**	2	Online	50	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: 1x OC-12 SONET, MM @ 2/0/*	**portInterfaceCard**	3	Online	41	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: 1x OC-12 SONET, MM @ 2/1/*	**portInterfaceCard**	3	Online	41	None	0:0:00:21.85	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	3	0	PIC: 1x OC-12 SONET, MM @ 2/2/*	**portInterfaceCard**	3	Online	41	–	0:0:00:21.85	0:0:00:00.00	–
8	3	4	0	PIC: @ 2/3/*	**portInterfaceCard**	3	Online	41	–	0:0:00:21.85	0:0:00:00.00	–
8	4	1	0	PIC: 1x OC-48 SONET, SMIR @ 3/0/*	**portInterfaceCard**	4	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	**portInterfaceCard**	4	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	**portInterfaceCard**	4	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	**portInterfaceCard**	4	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	**portInterfaceCard**	5	Online	43	–	0:0:00:21.87	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	**portInterfaceCard**	5	Online	43	–	0:0:00:21.87	0:0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	**portInterfaceCard**	5	Online	43	–	0:0:00:21.87	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	**portInterfaceCard**	5	Online	43	–	0:0:00:21.87	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	**portInterfaceCard**	6	Online	38	–	0:0:00:21.89	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	**portInterfaceCard**	6	Online	38	–	0:0:00:21.89	0:0:00:00.00	0
8	6	3	0	PIC: 1x OC-12 SONET, SMIR @ 5/2/*	**portInterfaceCard**	6	Online	38	–	0:0:00:21.89	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	6	4	0	PIC: 1x OC-12 SONET, MM @ 5/3/*	**portInterfaceCard**	6	Online	38	–	0:0:00:21.89	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	**portInterfaceCard**	7	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	**portInterfaceCard**	7	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	**portInterfaceCard**	7	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	**portInterfaceCard**	7	Empty	0	–	0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: 8x FE-FX, 100 BASE-FX @ 7/0/*	**portInterfaceCard**	8	Online	46	–	0:0:00:21.91	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	**portInterfaceCard**	8	Online	46	–	0:0:00:21.91	0:0:00:00.00	0
8	8	3	0	PIC: 1x Link Service(4) @ 7/2/*	**portInterfaceCard**	8	Online	46	–	0:0:00:21.91	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	**portInterfaceCard**	1	Online	46	–	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	**routingEngine**	2	Online	46	–	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	**routingEngine**	1	Present	34	–	0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM CMB	**frontPanelModule**	1	Online	28	–	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	**frontPanelModule**	1	Online	28	–	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	PCG 0	**clockGenerator**	1	Online	45	–	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	PCG 1	**clockGenerator**	2	Online	50	–	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 68: jnxFruContents Objects in the jnxFruTable of an M40e Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
12	1	0	0	MCS 0	**controlBoard**	1	Online	46	–	0:0:00:00.00	0:0:00:00.00	0
12	2	0	0	MCS 1	**controlBoard**	2	Online	0	–	0:0:00:00.00	0:0:00:00.00	0
13	1	0	0	CIP	**craftInterfacePanel**	1	Present	0	–	0:0:00:00.00	0:0:00:00.00	0

To verify L1, L2, and L3 indexes, use the **show chassis hardware** command. The sample output from an M40e router is as follows:

```
user@host> show chassis hardware
Item             Version  Part number  Serial number     Description
Chassis                                19084             M40e
Midplane         REV 01   710-005071   AX3654
FPM CMB          REV 03   710-001642   AR9037
FPM Display      REV 03   710-001647   AP1334
CIP              REV 08   710-001593   AE8486
PEM 0            Rev 01   740-003787   ME13120           Power Entry Module
PEM 1            Rev 01   740-003787   MC25354           Power Entry Module
PCG 0            REV 07   710-001568   AG1377
PCG 1            REV 07   710-001568   AR3806
Routing Engine 0 REV 04   740-003239   9001026568        RE-2.0
Routing Engine 1
MCS 0            REV 11   710-001226   AN5810
MCS 1            REV 11   710-001226   AR0109
SFM 1 SPP        REV 07   710-001228   BE0106
SFM 1 SPR        REV 05   710-002189   BE0062            Internet Processor II
FPC 0            REV 01   710-005078   BE0642            M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2496
  PIC 1          REV 04   750-001895   HE0885            1x OC-12 SONET, MM
  PIC 2          REV 06   750-003009   HE1422            4x CT3
  PIC 3          REV 03   750-003837   AP7134            1x Multi Link(32)
FPC 2            REV 01   710-005078   BE0647            M40e-FPC Type 1
  CPU            REV 01   710-004600   AN4299
  PIC 0          REV 04   750-001895   HD2623            1x OC-12 SONET, MM
  PIC 1          REV 04   750-001895   HE0609            1x OC-12 SONET, MM
  PIC 2          REV 04   750-001895   HE0871            1x OC-12 SONET, MM
FPC 3            REV 01   710-005197   BD9846            M40e-FPC Type 2
  CPU            REV 01   710-004600   BD2364
  PIC 0          REV 01   750-001900   AA9649            1x OC-48 SONET, SMIR
FPC 5            REV 01   710-005078   BE0639            M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2587
  PIC 2          REV 04   750-001896   AV4480            1x OC-12 SONET, SMIR
  PIC 3          REV 04   750-001895   HE1000            1x OC-12 SONET, MM
FPC 7            REV 01   710-005196   BD9456            M40e-FPC
  CPU            REV 01   710-004600   AN4323
  PIC 0          REV 01   750-004944   AY4645            8x FE-FX, 100 BASE-FX
  PIC 2          REV 01   750-007927   AP1919            1x Link Service(4)
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an M40e router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
```


Copyright © 2012, Juniper Networks, Inc.

```
    Current state          Master
    Election priority          Master (default)
    Temperature                          34 degrees C / 93 degrees F
    DRAM                                768 MB
    Memory utilization                   9 percent
    CPU utilization:
      User                        0 percent
      Background                         0 percent
      Kernel                      2 percent
      Interrupt                    0 percent
      Idle                               97 percent
    Model                          RE-2.0
    Serial ID                      9c000007c8644701
    Start time                     2002-06-24 10:33:41 PDT
    Uptime                         31 minutes, 7 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.01       0.02       0.00
Routing Engine status:
  Slot 1:
    Current state                  Present
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from an M40e router is as follows:

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            41      4          0        32         3         40
  1  Empty              0      0          0         0         0          0
  2  Online            43      4          0        32         1         40
  3  Online            38      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Online            46      4          0        32         1         40
  6  Empty              0      0          0         0         0          0
  7  Online            44      4          0        32         2         39
```

Table 69 on page 333 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of a T640 router.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	**power EntryModule**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	**power EntryModule**	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	217044
4	1	1	0	Top left front fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Top left middle fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top left rear fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	1	4	0	Top right front fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	5	0	Top right middle fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	6	0	Top right rear fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	6	0	Bottom right rear fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	1	0	Fourth blower from top	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	2	0	Bottom blower	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	3	3	0	Middle blower	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	4	0	Top blower	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	5	0	Second blower from top	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	**flexiblePic Concentrator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	1	0	FPC @ 0/0/* top temper-ature sensor	**flexiblePic Concentrator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	2	0	FPC @ 0/1/* bottom temper-ature sensor	**flexiblePic Concentrator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	**flexiblePic Concentrator**	2	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	1	0	FPC @ 1/0/* top temper-ature sensor	**flexiblePic Concentrator**	2	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	2	0	FPC @ 1/1/* bottom temper-ature sensor	**flexiblePic Concentrator**	2	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	**flexiblePic Concentrator**	3	Online	30	None	0:0:00:01.96	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	3	1	0	FPC @ 2/0/* top temperature sensor	**flexiblePic Concentrator**	3	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	3	2	0	FPC @ 2/1/* bottom temperature sensor	**flexiblePic Concentrator**	3	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	**flexiblePic Concentrator**	4	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	1	0	FPC @ 3/0/* top temperature sensor	**flexiblePic Concentrator**	4	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	2	0	FPC @ 3/1/* bottom temperature sensor	**flexiblePic Concentrator**	4	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	5	9	0	FPC @ 4/*/*	**flexiblePic Concentrator**	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	1	0	FPC @ 4/0/* top temperature sensor	**flexiblePic Concentrator**	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	2	0	FPC @ 4/1/* bottom temperature sensor	**flexiblePic Concentrator**	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	**flexiblePic Concentrator**	6	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	6	1	0	FPC @ 5/0/* top temperature sensor	**flexiblePic Concentrator**	6	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	2	0	FPC @ 5/1/* bottom temperature sensor	**flexiblePic Concentrator**	6	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	**flexiblePic Concentrator**	7	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	1	0	FPC @ 6/0/* top temperature sensor	**flexiblePic Concentrator**	7	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	2	0	FPC @ 6/1/* bottom temperature sensor	**flexiblePic Concentrator**	7	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	**flexiblePic Concentrator**	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	1	0	FPC @ 7/0/* top temperature sensor	**flexiblePic Concentrator**	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	2	0	FPC @ 7/1/* bottom temperature sensor	**flexiblePic Concentrator**	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	**portInterface-Card**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	2	0	PIC: @ 0/1/*	**portInterface-Card**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	**portInterface-Card**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	**portInterface-Card**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x OC-48 SONET, SMIR @ 1/0/*	**portInterface-Card**	2	Online		None	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 1x OC-48 SONET, SMSR @ 1/1/*	**portInterface-Card**	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 1x OC-48 SONET, SMIR @ 1/2/*	**portInterface-Card**	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x OC-48 SONET, SMIR @ 1/3/*	**portInterface-Card**	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	**portInterface-Card**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	**portInterface-Card**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	**portInterface-Card**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	**portInterface-Card**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	**portInterface-Card**	4	Online		None	0:0:00:01.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	**portInterface-Card**	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	4	3	0	PIC: @ 3/2/*	portInterface-Card	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterface-Card	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	5	1	0	PIC: 1x Tunnel @ 4/0/*	portInterface-Card	5	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	5	2	0	PIC: 1x OC-192 SM SR2 @ 4/1/*	portInterface-Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	5	3	0	PIC: 4x OC-48 SONET, SMSR @ 4/2/*	portInterface-Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	5	4	0	PIC: 1x OC-192 SM SR1 @ 4/3/*	portInterface-Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterface-Card	6	Empty	0	None	0:0:00:01.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterface-Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterface-Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterface-Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterface-Card	7	Online	30	None	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterface-Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterface-Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterface-Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	1	0	PIC: @ 7/0/*	**portInterface-Card**	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	**portInterface-Card**	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	**portInterface-Card**	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	**portInterface-Card**	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	**routing-Engine**	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	**routing-Engine**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM GBUS	**frontPanel-Module**	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	**frontPanel-Module**	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	**clockGenerator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	**clockGenerator**	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	**control-Board**	1	Online	27	None	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	**control-Board**	2	Unknown	0	None	0:0:00:01.96	0:0:00:00.00	0
13	1	0	0	CIP	**craftInter-facePanel**	1	Present	36	None	0:0:00:00.00	0:0:00:00.00	0
14	1	0	0	SPMB 0	**processor Mezzanine BoardForSIB**	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0
14	2	0	0	SPMB 1	**processor Mezzanine BoardForSIB**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 69: jnxFruEntry Objects in the jnxFruTable of a T640 Router *(continued)*

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
15	1	0	0	SIB 0	**switchInterfaceBoard**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	**switchInterfaceBoard**	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	**switchInterfaceBoard**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	**switchInterfaceBoard**	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	**switchInterfaceBoard**	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                1234              T640
Midplane         REV 04   710-002726   AX5603
FPM GBUS         REV 02   710-002901   HE3062
FPM Display      REV 01   710-002897   HD3033
CIP              REV 05   710-002895   HA5022
PEM 1            RevX02   740-002595   MD21812           Power Entry Module
SCG 1            REV 01   710-003423   HD3025
Routing Engine 0 REV 01   740-005022   210865700336      RE-3.0
CB 0             REV 02   710-002728   HE3025
CB 1
FPC 1            REV 01   710-002385   HE3173            FPC Type 2
  CPU            REV 06   710-001726   HC0042
  PIC 0          REV 03   750-001900   AD5737            1x OC-48 SONET, SMIR
  PIC 1          REV 07   750-001900   AR3613            1x OC-48 SONET, SMSR
  PIC 2          REV 01   750-001900   AA9604            1x OC-48 SONET, SMIR
  PIC 3          REV 01   750-001900   AA9602            1x OC-48 SONET, SMIR
  MMB 1          REV 03   710-001723   HC0111            MMB-144mbit
  ICBM           REV 04   710-003384   HA4497
  PPB 0          REV 02   710-003758   HA4543            PPB Type 2
  PPB 1          REV 02   710-003758   HA4540            PPB Type 2
FPC 2            REV 01   710-002385   HE3180            FPC Type 2
  CPU            REV 06   710-001726   HE7904
  MMB 1          REV 03   710-001723   HC0120            MMB-144mbit
  ICBM           REV 01   710-003384   HE3046
  PPB 0          REV 02   710-003758   HA4564            PPB Type 2
  PPB 1          REV 02   710-003758   HA4554            PPB Type 2
FPC 4            REV 04   710-001721   HE3145            FPC Type 3
  CPU            REV 06   710-001726   HC0034
  PIC 0                                                  1x Tunnel
  PIC 1          REV 01   750-003824   HE7803            1x OC-192 SM SR2
  PIC 2          REV 01   750-003336   HE3420            4x OC-48 SONET, SMSR
  PIC 3          REV 01   750-003824   HE7802            1x OC-192 SM SR1
```

```
  MMB 0          REV 03   710-001723   HE7230            MMB-144mbit
  MMB 1          REV 03   710-001723   HE7267            MMB-144mbit
  ICBM           REV 04   710-003384   HA4485
  PPB 0          REV 02   710-002845   HA4550            PPB Type 3
  PPB 1          REV 02   710-002845   HA4525            PPB Type 3
FPC 5            REV 04   710-001721   HE3175            FPC Type 3
  CPU
FPC 6            REV 01   710-002385   HD5027            FPC Type 2
  CPU            REV 06   710-001726   HC0033
  MMB 1          REV 03   710-001723   HC0080            MMB-144mbit
  ICBM           REV 04   710-003384   HA4486
  PPB 0          REV 02   710-003758   HA4541            PPB Type 2
  PPB 1          REV 02   710-003758   HA4539            PPB Type 2
SPMB 0           REV 01   710-003229   HA5999
SIB 0            REV 01   710-003980   HD5054            SIB-I8
SIB 2            REV 01   710-003980   HC0035            SIB-I8
SIB 3            REV 01   710-003980   HA5065            SIB-I8
SIB 4            REV 01   710-003980   HE3016            SIB-I8
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty              0      0          0         0         0          0
  1  Online            30      2          0       512         3         41
  2  Online            30      2          0       256         7         41
  3  Empty              0      0          0         0         0          0
  4  Online            30      4          0       512         6         41
  5  Offline         --- Unresponsive ---
  6  Online            30      2          0       256         7         41
  7  Empty              0      0          0         0         0          0
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                      2048 MB
    Memory utilization           4 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     2 percent
      Interrupt                  0 percent
      Idle                      97 percent
    Model                          RE-3.0
    Start time                     2002-06-24 10:33:34 PDT
    Uptime                         33 minutes, 38 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.08       0.03       0.01
```

To verify SPMB status, use the **show chassis spmb** command. Sample command output from a T640 router is as follows:

```
user@host> show chassis spmb
```


Copyright © 2012, Juniper Networks, Inc.

```
Slot 0 information:
  State                          Online
  Total CPU Utilization             2%
  Interrupt CPU Utilization         0%
  Memory Heap Utilization           0%
  Buffer Utilization               40%
  Start time:                    2002-06-24 10:34:22 PDT
  Uptime:                         33 minutes, 3 seconds
```

Table 70 on page 343 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of a T4000 router.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	**power EntryModule**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	**power EntryModule**	0	Online	27	None	0:0:00:00.00	0:0:00:00.00	217044
4	1	1	0	Front Top Fan Tray	**fan**	0	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	1	0	Top Left Front fan	**fan**	0	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Top left Middle fan	**fan**	0	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top left rear fan	**fan**	0	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top right rear fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Top right front fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	5	0	Top right middle fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	6	0	Top right rear fan	**fan**	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	2	0	0	Front Bottom Fan Tray	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	6	0	Bottom right rear fan	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear fan tray	**fan**	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	1	0	Rear Tray Top fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	2	0	Rear Tray Second fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	3	0	Rear Tray Third fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	3	4	0	Rear Tray Fourth fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	5	0	Rear Tray Fifth fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	6	0	Rear Tray Sixth fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	7	0	Rear Tray seventh fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	8	0	Rear Tray Bottom fan	**fan**	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	3	0	ST-MMB2 @ 0/*/* MMB0	**flexiblePic Concentrator**	0	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	4	0	FPC @ 0/*/* MMB1	**flexiblePic Concentrator**	0	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC Type 4-ES @ 1/*/*	**flexiblePic Concentrator**	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	1	0	FPC Type 4-ES @ 1/*/* top temperature	**flexiblePic Concentrator**	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	2	0	FPC Type 4-ES @ 1/*/* bottom temperature	**flexiblePic Concentrator**	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	2	3	0	ST-MMB2 @ 1/*/* MMB0	flexiblePic Concentrator	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	4	0	ST-MMB2 @ 1/*/* MMB1	flexiblePic Concentrator	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	3	0	0	FPC Type 5-3D @ 2/*/*	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	3	1	0	FPC Type 5-3D @ 2/*/* intake	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	3	2	0	FPC Type 5-3D @ 2/*/* exhaust	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	3	3	0	FPC @ 2/*/* MMB0	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	3	4	0	FPC @ 2/*/* MMB1	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePic Concentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	1	0	FPC @ 3/*/* top temperature	flexiblePic Concentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	2	0	FPC @ 3/*/* bottom temperature	flexiblePic Concentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	3	0	FPC @ 3/*/* MMB0	flexiblePic Concentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	4	4	0	FPC @ 3/*/* MMB1	**flexiblePic Concentrator**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC Type 5-3D @ 4/*/*	**flexiblePic Concentrator**	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	1	0	FPC Type 5-3D @ 4/*/* intake	**flexiblePic Concentrator**	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	2	0	FPC Type 5-3D @ 4/*/* exhaust	**flexiblePic Concentrator**	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	3	0	FPC @ 4/*/* MMB0	**flexiblePic Concentrator**	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	5	4	0	FPC @ 4/*/* MMB1	**flexiblePic Concentrator**	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	**flexiblePic Concentrator**	5	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	1	0	FPC @ 5/*/* top temperature sensor	**flexiblePic Concentrator**	5	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	2	0	FPC @ 5/*/* bottom temperature sensor	**flexiblePic Concentrator**	5	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	3	0	FPC @ 5/*/* MMB0	**flexiblePic Concentrator**	5	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	2	0	FPC @ 5/*/* MMB1	**flexiblePic Concentrator**	5	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	7	0	0	FPC @ 6/*/*	**flexiblePic Concentrator**	6	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	1	0	FPC @ 6/0/* top temperature sensor	**flexiblePic Concentrator**	6	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	2	0	FPC @ 6/1/* bottom temperature sensor	**flexiblePic Concentrator**	6	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	3	0	FPC @ 6/*/* MMB0	**flexiblePic Concentrator**	6	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	7	4	0	FPC @ 6/*/* MMB1	**flexiblePic Concentrator**	6	Online	30	None	0:0:00:02.05	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	1	0	FPC @ 7/*/* top temperature sensor	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	2	0	FPC @ 7/*/* bottom temperature sensor	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	3	0	FPC @ 7/*/* MMB0	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	8	4	0	FPC @ 7/*/* MMB1	**flexiblePic Concentrator**	7	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	**portInterface-Card**	0	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	**portInterface-Card**	1	Online		None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	2	0	PIC: @ 1/1/*	**portInterface-Card**	1	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	**portInterface-Card**	1	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	**portInterface-Card**	1	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/0/*	**portInterface-Card**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/1/*	**portInterface-Card**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	**portInterface-Card**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	**portInterface-Card**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	**portInterface-Card**	3	Online		None	0:0:00:01.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	**portInterface-Card**	3	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	**portInterface-Card**	3	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*PIC: @ 3/3/*	**portInterface-Card**	3	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	5	1	0	PIC: 24x10GE (LAN/WAN) SFPP @ 4/0/*	**portInterface-Card**	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	2	0	PIC: 12x10GE (LAN/WAN) SFPP @ 4/1/*	**portInterface-Card**	4	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	**portInterface-Card**	4	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	**portInterface-Card**	5	Empty	0	None	0:0:00:01.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	**portInterface-Card**	5	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	**portInterface-Card**	5	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	**portInterface-Card**	5	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	**portInterface-Card**	6	Online	30	None	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	**portInterface-Card**	6	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	**portInterface-Card**	6	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	**portInterface-Card**	6	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	**portInterface-Card**	7	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	**portInterface-Card**	7	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	**portInterface-Card**	7	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	**portInterface-Card**	7	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	**routing-Engine**	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

Table 70: jnxFruEntry Objects in the jnxFruTable of a T4000 Router *(continued)*

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
9	2	0	0	Routing Engine 1	**routing-Engine**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM GBUS	**frontPanel-Module**	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	**frontPanel-Module**	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	**clockGenerator**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	**clockGenerator**	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	**control-Board**	1	Online	27	None	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	**control-Board**	2	Unknown	0	None	0:0:00:01.96	0:0:00:00.00	0
13	1	0	0	CIP	**craftInter-facePanel**	1	Present	36	None	0:0:00:00.00	0:0:00:00.00	0
14	1	0	0	SPMB 0	**processor Mezzanine BoardForSIB**	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0
14	2	0	0	SPMB 1	**processor Mezzanine BoardForSIB**	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	1	0	0	SIB 0	**switchInter-faceBoard**	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	**switchInter-faceBoard**	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	**switchInter-faceBoard**	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	**switchInter-faceBoard**	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	**switchInter-faceBoard**	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0

Copyright © 2012, Juniper Networks, Inc.

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T4000 router is as follows:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JN1172F25AHA      T4000
Midplane         REV 01   710-027486   RC8355            T-series Backplane
FPM GBUS         REV 13   710-002901   BBAE0927          T640 FPM Board
FPM Display      REV 01   710-021387   EF6764            T1600 FPM Display
CIP              REV 06   710-002895   BBAD9210          T-series CIP
PEM 0            REV 01   740-036442   VA00016           Power Entry Module 6x60
SCG 0            REV 18   710-003423   BBAD7248          T640 Sonet Clock Gen.
SCG 1            REV 18   710-003423   BBAE3874          T640 Sonet Clock Gen.
Routing Engine 0 REV 05   740-026941   P737F-002248      RE-DUO-1800
Routing Engine 1 REV 06   740-026941   P737F-002653      RE-DUO-1800
CB 0             REV 09   710-022597   ED0295            LCC Control Board
CB 1             REV 09   710-022597   EA6050            LCC Control Board
FPC 0            REV 26   750-032819   EK1173            FPC Type 5-3D
  CPU            REV 12   711-030686   EJ8584            SNG PMB
  PIC 0          REV 07   750-034624   EF6837            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   123363A01145      SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   123363A01147      SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01P3           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B10M03256         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   AJJ01M2           SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   123363A01137      SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01PN           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   AJJ01NW           SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   123363A01139      SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJJ01KE           SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   123363A01336      SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B10M01325         SFP+-10G-SR
  PIC 1          REV 07   750-034624   EF6800            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   AJJ01SA           SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   AJJ01QZ           SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJH0217           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   AJJ01TE           SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   AJJ01KV           SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   AJJ01MU           SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01R0           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   AJJ01TC           SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   AJJ0364           SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJD0GV3           SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   B10M03343         SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   AJJ01QJ           SFP+-10G-SR
  LMB 0          REV 05   711-034381   EJ8490            Type-0 LMB
  LMB 1          REV 04   711-035774   EJ8517            Type-1 LMB
  LMB 2          REV 05   711-034381   EJ8489            Type-0 LMB
FPC 3            REV 07   750-032819   EG3637            FPC Type 5-3D
  CPU            REV 09   711-030686   EG0150            SNG PMB
  PIC 0          REV 08   750-035293   EF3657            1x100GE
    Xcvr 0       REV 01   740-032210   C22CQNJ           CFP-100G-LR4
  PIC 1          REV 10   750-034624   BBAN4098          12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   B11J04902         SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   B11J04891         SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01MX           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B11J04183         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B11J04894         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   B11J04184         SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   B11J04897         SFP+-10G-SR
```


Copyright © 2012, Juniper Networks, Inc.

```
    Xcvr 7       REV 01   740-031980   B11J04899         SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   AJJ01TV           SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   B11J04057         SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   AJJ01M4           SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B11J04905         SFP+-10G-SR
  LMB 0          REV 04   711-034381   EG1524            Type-0 LMB
  LMB 1          REV 03   711-035774   EG0345            Type-1 LMB
  LMB 2          REV 04   711-034381   EG1522            Type-0 LMB
FPC 5            REV 03   710-033871   BBAJ0768          FPC Type 4-ES
  CPU            REV 11   710-016744   BBAH9342          ST-PMB2
  PIC 0          REV 09   750-029262   EE6789            100GE
  PIC 1          REV 03   750-034781   EE6655            100GE CFP
    Xcvr 0       REV 01   740-032210   J11A22334         CFP-100G-LR4
  BRIDGE 0       REV 03   711-029995   EE6572            100GE Bridge Board
  MMB 0          REV 07   710-025563   BBAJ4657          ST-MMB2
  MMB 1          REV 07   710-025563   BBAJ3073          ST-MMB2
FPC 6            REV 05   750-010153   EF4936            FPC Type 5-3D
  CPU            REV 06   711-030686   EF4189            SNG PMB
  PIC 0          REV 10   750-034624   BBAN4109          12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   B11J04895         SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   B11J04898         SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   B11J04021         SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   B11J04903         SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B11J04311         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   B11J04059         SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   B11J04016         SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   B11J04017         SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   B11J04887         SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   B11J04297         SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   B11J04893         SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   B11J04022         SFP+-10G-SR
  PIC 1          REV 02   750-034624   EE3711            12x10GE (LAN/WAN) SFPP
    Xcvr 0       REV 01   740-031980   AJH033X           SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   AJJ01N0           SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AJJ01SV           SFP+-10G-SR
    Xcvr 3       REV 01   740-031980   AJJ032L           SFP+-10G-SR
    Xcvr 4       REV 01   740-031980   B10M01593         SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   AJD0FF1           SFP+-10G-SR
    Xcvr 6       REV 01   740-031980   AJJ01NU           SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   123363A01305      SFP+-10G-SR
    Xcvr 8       REV 01   740-031980   B10M00361         SFP+-10G-SR
    Xcvr 9       REV 01   740-031980   AJJ01M7           SFP+-10G-SR
    Xcvr 10      REV 01   740-031980   AJJ032X           SFP+-10G-SR
    Xcvr 11      REV 01   740-031980   AJJ01PG           SFP+-10G-SR
  LMB 0          REV 04   711-034381   EF3838            Type-0 LMB
  LMB 1          REV 03   711-035774   EF3821            Type-1 LMB
  LMB 2          REV 04   711-034381   EF3834            Type-0 LMB
SPMB 0           REV 05   710-023321   ED1990            LCC Switch CPU
SPMB 1           REV 05   710-023321   EA2768            LCC Switch CPU
SIB 0            REV 02   711-036340   EF8802            SIB-HC-3D
SIB 1            REV 07   711-036340   EG2286            SIB-HC-3D
SIB 2            REV 07   711-036340   EG2252            SIB-HC-3D
SIB 3            REV 02   711-036340   EF1358            SIB-HC-3D
SIB 4            REV 02   711-036340   EF8806            SIB-HC-3D
Fan Tray 0                                               Front Top Fan Tray
Fan Tray 1                                               Front Bottom Fan Tray
-- Rev 2
Fan Tray 2                                               Rear Fan Tray -- Rev 3
```

To verify FPC status, use the **show chassis fpc** command. Sample command output from a T4000 router is as follows:

```
user@host> show chassis fpc

regress@stymphalian# run show chassis fpc
                      Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State             (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online             48     15          0       2816       21         27
  1  Empty
  2  Empty
  3  Online             51     15          0       2816       21         27
  4  Empty
  5  Online             39      8          0       2048        6         23
  6  Online             49     15          0       2816       21         27
  7  Empty
```

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from a T4000 router is as follows:

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 33 degrees C / 91 degrees F
    CPU temperature             50 degrees C / 122 degrees F
    DRAM                      8960 MB
    Memory utilization          18 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     4 percent
      Interrupt                  1 percent
      Idle                      95 percent
    Model                          RE-DUO-1800
    Serial ID                      P737F-002248
    Start time                     2012-02-09 22:49:53 PST
    Uptime                         2 hours, 21 minutes, 35 seconds
    Last reboot reason             Router rebooted after a normal shutdown.
    Load averages:                 1 minute   5 minute  15 minute
                                       0.00       0.04       0.00
Routing Engine status:
  Slot 1:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                 32 degrees C / 89 degrees F
    CPU temperature             46 degrees C / 114 degrees F
    DRAM                      8960 MB
    Memory utilization          24 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     0 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                          RE-DUO-1800
    Serial ID                      P737F-002653
    Start time                     2012-02-08 20:12:51 PST
    Uptime                         1 day, 4 hours, 58 minutes, 28 seconds
    Last reboot reason             Router rebooted after a normal shutdown.
```

To verify SPMB status, use the **show chassis spmb** command. Sample command output from a T4000 router is as follows:

 Copyright © 2012, Juniper Networks, Inc.

```
user@host> show chassis spmb

Slot 0 information:
  State                          Online
  Total CPU Utilization            18%
  Interrupt CPU Utilization         0%
  Memory Heap Utilization           0%
  Buffer Utilization               22%
  Start time:                    2012-02-09 22:51:09 PST
  Uptime:                        2 hours, 25 minutes, 45 seconds
Slot 1 information:
  State                          Online - Standby
  Total CPU Utilization             0%
  Interrupt CPU Utilization         0%
  Memory Heap Utilization           0%
  Buffer Utilization               22%
  Start time:                    2012-02-09 22:51:10 PST
  Uptime:                        2 hours, 25 minutes, 44 seconds
```

jnxBoxKernelMemoryUsedPercent

The object identifier for **jnxBoxKernelMemoryUsedPercent** is **{jnxBoxAnatomy 16}**. This object represents the amount of the kernel memory used, expressed as a percentage of the total available memory. The object shows **0** if the kernel memory usage is unavailable or inapplicable. When the kernel memory usage exceeds 80 percent, a system log message is logged and an RMON rising threshold trap is generated if RMON health monitoring is enabled for the device.

jnxBoxSystemDomainType

The object identifier for **jnxBoxSystemDomainType** is **{jnxBoxAnatomy 17}**. This object indicates the domain type of the device, that is, whether it is a root system domain (RSD; represented by integer 2) or a protected system domain (PSD; represented by integer 3). This object returns an integer value of **1**, denoting not applicable, if the system domain type feature is not supported on the device.

Related Documentation

- Chassis MIBs
- Chassis MIB Textual Conventions on page 240
- Chassis Traps on page 355

Chassis Traps

The chassis-related traps are defined under the **jnxTraps** and **jnxChassisOKtraps** branches. For the system logging severity levels for these traps, see ."Juniper Networks Enterprise-Specific SNMP Traps" on page 125.

These traps are defined as follows:

- Power failure (**jnxPowerSupplyFailure**)—Sent when the power supply, router circuit breaker, or power circuit fails, or when there is a power outage. When only one of the power supplies has failed, the service impact is minimal. One power supply can provide the necessary power for a fully loaded router. This trap is repeated every hour until the power supply is restored.

  To determine the source of the failure, you must physically inspect the router. For a router with six-input DC power supply, you must also inspect the **feeds** configuration. For six-input DC power supply, the **jnxPowerSupplyFailure** trap is sent when the number of input feeds connected to the power supply is fewer than the value configured for the **feeds** statement at the **[edit chassis pem]** or **[edit chassis lcc *lcc-number* pem]** hierarchy level.

- Fan failure (**jnxFanFailure**)—Sent when the fan fuse blows or when the fan wiring shorts out. When only one fan has failed, there is no service impact. The remaining fans increase speed to compensate. However, you must resolve the problem before another fan fails. This trap is repeated every hour until the fan failure is fixed. To determine the source of the failure, you must physically inspect the router, taking care to check the fuses. See the hardware installation guide for your router model for more information.

- Overtemperature (**jnxOverTemperature**)—Sent when several fans fail or the room temperature increases significantly. The service impact of this trap depends on the temperature of the router. In general, the router increases the speed of the fans when any component exceeds a temperature of 55º C. The fans remain at the higher speed until the temperature decreases below the threshold. In this case, there is no service impact. However, if the temperature exceeds 75º C, the router transmits a warning and automatically shuts down. This scenario creates a significant service impact because the shutdown affects additional routers and equipment. This trap is repeated every minute until the temperature is brought down to normal. To determine the source of the overtemperature problem, you must physically inspect the router to determine whether any fans have failed in the router.

- Power Supply OK (**jnxPowerSupplyOK**)—Sent when a power supply recovers from failure.

  The six-input DC power supply fails when the number of input feeds connected to the power supply is fewer than the value configured for the **feeds** statement at the **[edit chassis pem]** or **[edit chassis lcc *lcc-number* pem]** hierarchy level. Therefore, for a failed six-input DC power supply, this trap is sent when one of the following events occurs:

  - The input feeds connected to the six-input DC power supply are energized in such a way that the number of input feeds connected to the power supply is equal to or greater than the value configured for the **feeds** statement.
  - The **feeds** statement is configured with a value that is equal to or less than the number of input feeds connected to the six-input DC power supply.

- Fan OK (**jnxFanOK**)—Sent when a fan recovers from failure.

- Temperature OK (**jnxTemperatureOK**)—Sent when a chassis component recovers from an overtemperature condition.

 Copyright © 2012, Juniper Networks, Inc.

- Redundancy Switchover (**jnxRedundancySwitchover**)—For certain platforms, such as the M20 or M160, some subsystems, such as the Routing Engine, have a redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions. The redundancy switchover trap indicates such a change.
- Field Replaceable Unit Removal (**jnxFruRemoval**)—Sent when the specified FRU has been removed from the chassis.
- Field Replaceable Unit Insertion (**jnxFruInsertion**)—Sent when the specified FRU has been inserted into the chassis.
- Field Replaceable Unit Power Off (**jnxFruPowerOff**)—Sent when the specified FRU has been powered off in the chassis.

  The **jnxFruPowerOff** trap is also sent in the following scenarios:

  - When an FRU that is controlled using inter-process communication (IPC) goes offline or is removed from the chassis. For example, a switch interface board (SIB).
  - When an FRU that does not have a backup unit goes offline or is removed from the chassis. For example, a Flexible PIC Concentrator (FPC).

  NOTE: **When a SONET Clock Generator (SCG) is taken offline, the unit is not powered down. Therefore, jnxFruPowerOff or jnxFruPowerOn traps are not sent when the unit is taken online or offline.**

- Field Replaceable Unit Power On (**jnxFruPowerOn**)—Sent when the specified FRU has been powered on in the chassis.
- Field Replaceable Unit Failed (**jnxFruFailed**)—Sent when the specified FRU has failed in the chassis. Typically, this is due to the FRU not powering up or being unable to load software. FRU replacement might be required.
- Field Replaceable Unit Offline (**jnxFruOffline**)—Sent when the specified FRU goes offline. However, when an FRU that does not have a backup unit goes offline, the Junos OS generates the **jnxFruPowerOff** trap instead of the **jnxFruOffline** trap. Typically, a **jnxFruOffline** trap is generated to inform the backup FRU about the status of the primary FRU so that the backup FRU can take over when the primary FRU goes offline.

  The following are some scenarios when **jnxFruOffline** traps are generated:

  - When a Packet Forwarding Engine Clock Generator (PCG) goes offline (M40e).
  - When a SONET Clock Generator goes offline (T Series).
  - When a Line Card Chassis goes offline (TX4 internet router).
  - When a Routing Engine goes offline

- Field Replaceable Unit Online (**jnxFruOnline**)—Sent when the specified FRU goes online.
- Field Replaceable Unit Check (**jnxFruCheck**)—Sent when the specified FRU has encountered operational errors. On M120 and M320 routers, this trap is sent if the revision number for the ATM2 PIC FPGA is less than 8B44(4).

- Forwarding Engine Board (FEB) Switch Over (**jnxFEBSwitchover**)—Sent when the specified FEB undergoes a switchover.
- Hard disk Failure (**jnxHardDiskFailed**)—Sent when the hard disk in the specified Routing Engine is not operational.
- Hard disk Missing (**jnxHardDiskMissing**)—Sent when the hard disk in the specified Routing Engine is missing from the boot device list.

This section contains the following topics:

- SNMPv1 Trap Format on page 358
- SNMPv2 Trap Format on page 359

SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in Table 71 on page 358. To view the SNMPv1 chassis-related traps, see "Juniper Networks Enterprise-Specific SNMP Version 1 Traps" on page 126 .

The column headings describe the SNMPv1 traps format:

- Trap Name—The name of the trap.
- Enterprise ID—The identification number of the enterprise-specific trap.
- Generic Trap Number—The generic trap number field of the SNMP trap PDU. This field is **enterpriseSpecific(6)** for enterprise-specific traps, and other predefined values for standard traps.
- Specific Trap Number— The specific trap number field of the SNMP trap PDU. For standard traps, this field is zero; for enterprise-specific traps, this field is nonzero as defined in the enterprise-specific MIBs.

Table 71: SNMP Version 1 Trap Format

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2
jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2
jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12
jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9
jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6
jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10
jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7

Copyright © 2012, Juniper Networks, Inc.

Table 71: SNMP Version 1 Trap Format *(continued)*

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8
jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5
jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2	6	1
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1	6	4
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3

SNMPv2 Trap Format

The SNMPv2 trap format for the Chassis MIB traps is described in Table 72 on page 359.

The column headings describe the SNMPv2 traps format:

- Trap Name—The name of the trap.
- **snmpTrapOID**—The authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2 trap PDU and InformRequest PDU.
- Description—The Junos OS enterprise-specific name of the trap.

Table 72: SNMP Version 2 Trap Format

Trap Name	snmpTrapOID	Description
jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	The fan fuse has blown or the fan wiring has shorted out. This trap is generated every hour until the fan failure is fixed.
jnxFanOK	1.3.6.1.4.1.2636.4.2.2	The fan has recovered from a failure state.
jnxFruCheck	1.3.6.1.4.1.2636.4.1.12	The FRU has operational errors and has gone into a self-check diagnostic state. The revision number for the ATM2 PIC FPGA on an M120 or M320 router is less than 8B44(4).
jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	The FRU has been inserted into the chassis.
jnxFruFailed	1.3.6.1.4.1.2636.4.1.9	The FRU has failed in the chassis.

Copyright © 2012, Juniper Networks, Inc.

Table 72: SNMP Version 2 Trap Format *(continued)*

Trap Name	snmpTrapOID	Description
jnxFruOffline	1.3.6.1.4.1.2636.4.1.10	The FRU has gone offline.
jnxFruOnline	1.3.6.1.4.1.2636.4.1.11	The FRU has gone back online.
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	The FRU has been powered off in the chassis.
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	The FRU has been powered on in the chassis.
jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	The FRU has been removed from the chassis.
jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	Several fans have failed or the room temperature has increased significantly. This trap is repeated every minute until the temperature is brought down to normal.
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	The power supply, router circuit breaker, or power circuit failed, or there has been a power outage. This trap is generated every hour until the power supply is restored.
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	The power supply has recovered from a failure.
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1.4	A redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions.
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	The component sensor has detected an overtemperature condition.

Related Documentation

- Chassis MIBs
- Chassis MIB Textual Conventions on page 240
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Chassis Definitions for Router Model MIB

The enterprise-specific Chassis Definitions for Router Model MIB contain the OIDs that are used by the Chassis MIB to identify platform and chassis components. The Chassis

Copyright © 2012, Juniper Networks, Inc.

MIB provides information that changes often. The Chassis Definitions for the Router Model MIB provide information that changes less often.

The last number in each **sysObjectId**, shown in Table 73 on page 361, corresponds to the devices and therefore does not change.

Table 73: Router Model and Their sysObjectIds

Model	SysObjectID	jnxProductName
SRX1400	1.3.6.1.4.1.2636.1.1.1.2.49	**jnxProductNameSRX1400**
SRX3400	1.3.6.1.4.1.2636.1.1.1.2.35	**jnxProductNameSRX3400**
SRX3600	1.3.6.1.4.1.2636.1.1.1.2.34	**jnxProductNameSRX3600**

For a downloadable version of the Chassis Definitions for Router Model MIB, see http://www.juniper.net/techpubs/en_US/junos11.2/topics/reference/mibs/mib-jnx-chas-defines.txt .

Related Documentation

- Chassis MIB Textual Conventions on page 240
- Chassis MIBs

MIB Objects for the SRX1400 Services Gateway

The Chassis MIB objects for the SRX1400 Services Gateway include:

```
 jnxProductLineSRX1400      OBJECT IDENTIFIER ::= { jnxProductLine  49 }
 jnxProductNameSRX1400      OBJECT IDENTIFIER ::= { jnxProductName  49 }
 jnxProductModelSRX1400     OBJECT IDENTIFIER ::= { jnxProductModel  49 }
 jnxProductVariationSRX1400 OBJECT IDENTIFIER ::= { jnxProductVariation 49 }
 jnxChassisSRX1400          OBJECT IDENTIFIER ::= { jnxChassis  49 }

 jnxSlotSRX1400             OBJECT IDENTIFIER ::= { jnxSlot 49 }

   jnxSRX1400SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX1400 1  }
   jnxSRX1400SlotHM         OBJECT IDENTIFIER ::= { jnxSlotSRX1400 2  }
   jnxSRX1400SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX1400 3  }
   jnxSRX1400SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX1400 4  }
   jnxSRX1400SlotCB         OBJECT IDENTIFIER ::= { jnxSlotSRX1400 5  }
   jnxSRX1400SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotSRX1400 6  }

jnxMediaCardSpaceSRX1400    OBJECT IDENTIFIER ::= { jnxMediaCardSpace  49 }
jnxSRX1400MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX1400 1}


jnxMidplaneSRX1400          OBJECT IDENTIFIER ::= { jnxBackplane 49 }
```

Related Documentation

- Chassis MIBs
- Chassis MIB Textual Conventions on page 240

- MIB Objects for the SRX3400 Services Gateway on page 362
- MIB Objects for the SRX3600 Services Gateway on page 363

- MIB Objects for the SRX5600 Services Gateway
- MIB Objects for the SRX5800 Services Gateway
- MIB Objects for the SRX100 Services Gateway
- MIB Objects for the SRX210 Services Gateway
- MIB Objects for the SRX240 Services Gateway
- MIB Objects for the SRX550 Services Gateway
- MIB Objects for the SRX650 Services Gateway

MIB Objects for the SRX3400 Services Gateway

The Chassis MIB objects for the SRX3400 Services Gateway include:

```
  jnxProductLineSRX3400      OBJECT IDENTIFIER ::= { jnxProductLine  35 }
  jnxProductNameSRX3400      OBJECT IDENTIFIER ::= { jnxProductName  35 }
  jnxProductModelSRX3400     OBJECT IDENTIFIER ::= { jnxProductModel  35 }
  jnxProductVariationSRX3400 OBJECT IDENTIFIER ::= { jnxProductVariation 35 }
  jnxChassisSRX3400          OBJECT IDENTIFIER ::= { jnxChassis  35 }

  jnxSlotSRX3400             OBJECT IDENTIFIER ::= { jnxSlot 35 }

    jnxSRX3400SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX3400 1  }
    jnxSRX3400SlotHM         OBJECT IDENTIFIER ::= { jnxSlotSRX3400 2  }
    jnxSRX3400SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX3400 3  }
    jnxSRX3400SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX3400 4  }
    jnxSRX3400SlotCB         OBJECT IDENTIFIER ::= { jnxSlotSRX3400 5  }
    jnxSRX3400SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotSRX3400 6  }

 jnxMediaCardSpaceSRX3400    OBJECT IDENTIFIER ::= { jnxMediaCardSpace  35 }
 jnxSRX3400MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX3400 1}


 jnxMidplaneSRX3400          OBJECT IDENTIFIER ::= { jnxBackplane 35 }
```

Related Documentation

- Chassis MIBs
- Chassis MIB Textual Conventions on page 240
- MIB Objects for the SRX1400 Services Gateway on page 361
- MIB Objects for the SRX3600 Services Gateway on page 363
- MIB Objects for the SRX5600 Services Gateway
- MIB Objects for the SRX5800 Services Gateway
- MIB Objects for the SRX100 Services Gateway
- MIB Objects for the SRX210 Services Gateway

Copyright © 2012, Juniper Networks, Inc.

- MIB Objects for the SRX240 Services Gateway
- MIB Objects for the SRX550 Services Gateway
- MIB Objects for the SRX650 Services Gateway

MIB Objects for the SRX3600 Services Gateway

The Chassis MIB objects for the SRX3600 Services Gateway include:

```
  jnxProductLineSRX3600       OBJECT IDENTIFIER ::= { jnxProductLine  34 }
  jnxProductNameSRX3600       OBJECT IDENTIFIER ::= { jnxProductName  34 }
  jnxProductModelSRX3600      OBJECT IDENTIFIER ::= { jnxProductModel  34 }
  jnxProductVariationSRX3600  OBJECT IDENTIFIER ::= { jnxProductVariation 34 }
  jnxChassisSRX3600           OBJECT IDENTIFIER ::= { jnxChassis  34 }

   jnxSlotSRX3600             OBJECT IDENTIFIER ::= { jnxSlot 34 }
    jnxSRX3600SlotFPC         OBJECT IDENTIFIER ::= { jnxSlotSRX3600 1 }
    jnxSRX3600SlotHM          OBJECT IDENTIFIER ::= { jnxSlotSRX3600 2 }
    jnxSRX3600SlotPower       OBJECT IDENTIFIER ::= { jnxSlotSRX3600 3 }
    jnxSRX3600SlotFan         OBJECT IDENTIFIER ::= { jnxSlotSRX3600 4 }
    jnxSRX3600SlotCB          OBJECT IDENTIFIER ::= { jnxSlotSRX3600 5 }
    jnxSRX3600SlotFPB         OBJECT IDENTIFIER ::= { jnxSlotSRX3600 6 }

  jnxMediaCardSpaceSRX3600    OBJECT IDENTIFIER ::= { jnxMediaCardSpace  34 }
  jnxSRX3600MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX3600
1}

  jnxMidplaneSRX3600          OBJECT IDENTIFIER ::= { jnxBackplane  34 }
```

Related Documentation

- Chassis MIBs
- Chassis MIB Textual Conventions on page 240
- MIB Objects for the SRX3400 Services Gateway on page 362
- MIB Objects for the SRX5600 Services Gateway
- MIB Objects for the SRX5800 Services Gateway
- MIB Objects for the SRX100 Services Gateway
- MIB Objects for the SRX210 Services Gateway
- MIB Objects for the SRX240 Services Gateway
- MIB Objects for the SRX550 Services Gateway
- MIB Objects for the SRX650 Services Gateway

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 21

Interpreting the Enterprise-Specific Chassis Cluster MIBs

- Chassis Cluster MIB on page 365
- jnxJsChassisClusterSwitchover on page 365
- jnxJsChClusterIntfTrap on page 366
- Chassis Cluster Notifications on page 367

Chassis Cluster MIB

The Juniper Networks enterprise-specific chassis cluster MIB, whose object identifier is **{jnxJsChassisCluster 1}**, contains 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/junos12.1/topics/reference/mibs/mib-jnx-jsrpd.txt.

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

- jnxJsChassisClusterSwitchover on page 365
- jnxJsChClusterIntfTrap on page 366

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsChassisClusterSwitchover

The **jnxJsChassisClusterSwitchover** trap, whose object identifier is **{jnxJsChassisClusterNotifications 1}**, contains notification information about signal switchover or failover.

Table 74 on page 366 lists the objects associated with **jnxJsChassisClusterSwitchover**.

Table 74: Chassis Cluster Switchover Objects

Object	Object ID	Description
jnxJsChClusterSwitchoverInfoRedundancyGroup	jnxJsChassisClusterTrapObjects 1	Identification of the redundancy group that switched over.
jnxJsChClusterSwitchoverInfoClusterId	jnxJsChassisClusterTrapObjects 2	Identification number of the cluster.
jnxJsChClusterSwitchoverInfoNodeId	jnxJsChassisClusterTrapObjects 3	Node identification information where the switchover occurred.
jnxJsChClusterSwitchoverInfoPreviousState	jnxJsChassisClusterTrapObjects 4	Redundancy state of the cluster before the switchover.
jnxJsChClusterSwitchoverInfoCurrentState	jnxJsChassisClusterTrapObjects 5	Redundancy state of the cluster after the switchover.
jnxJsChClusterSwitchoverInfoReason	jnxJsChassisClusterTrapObjects 6	Reason for the switchover.

Related Documentation

- Chassis Cluster MIB on page 365
- jnxJsChClusterIntfTrap on page 366
- Chassis Cluster Notifications on page 367

jnxJsChClusterIntfTrap

The SNMP trap **jnxJsChClusterIntfTrap** trap, whose object identifier is **jnxJsChassisClusterNotifications 2**, is triggered whenever the state of the control link interfaces or fabric link interfaces change (up to down or down to up) in a chassis cluster deployment. This trap, including the following objects, is defined in the MIB (jnx-jsrpd.mib):

Table 75 on page 366 lists the objects associated with **jnxJsChClusterIntfTrap**.

Table 75: Chassis Cluster Trap Objects

Object	Object ID	Description
jnxJsChClusterSwitchoverInfoClusterId	jnxJsChassisClusterTrapObjects 2	Identification number of the cluster.
jnxJsChClusterIntfName	jnxJsChassisClusterTrapObjects 7	Name of the interface for which the state changed.
jnxJsChClusterIntfState	jnxJsChassisClusterTrapObjects 8	New state of the interface (up or down).

Copyright © 2012, Juniper Networks, Inc.

Table 75: Chassis Cluster Trap Objects *(continued)*

Object	Object ID	Description
jnxJsChClusterIntfSeverity	**jnxJsChassisClusterTrapObjects 9**	Severity of the change (minor or major). For example, in a dual control link or dual fabric link configuration, if one of the control or fabric links goes down, the event is considered to be minor. If the other control or fabric link also goes down, the event is considered to be major.
jnxJsChClusterIntfStateReason	**jnxJsChassisClusterTrapObjects 10**	Reason for the state change.

Related Documentation

- Chassis Cluster MIB on page 365
- jnxJsChassisClusterSwitchover on page 365
- Chassis Cluster Notifications on page 367

Chassis Cluster Notifications

The Junos OS generates the following traps whenever a chassis switchover occurs in a chassis cluster deployment:

- The **jnxJsChassisClusterSwitchover** trap, whose object identifier is **{jnxJsChassisClusterNotifications 1}**, contains **jnxJsChClusterSwitchoverInfoRedundancyGroup**, **jnxJsChClusterSwitchoverInfoClusterId**, **jnxJsChClusterSwitchoverInfoNodeId**, **jnxJsChClusterSwitchoverInfoPreviousState**, **jnxJsChClusterSwitchoverInfoCurrentState**, and **jnxJsChClusterSwitchoverInfoReason**.
- The **jnxJsChClusterIntfTrap** trap, whose object identifier is **{jnxJsChassisClusterNotifications 2}**, contains **jnxJsChClusterSwitchoverInfoClusterId**, **jnxJsChClusterIntfName**, **jnxJsChClusterIntfState**, **jnxJsChClusterIntfSeverity**, and **jnxJsChClusterIntfStateReason**.

Related Documentation

- Chassis Cluster MIB on page 365
- jnxJsChassisClusterSwitchover on page 365
- jnxJsChClusterIntfTrap on page 366

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 22

Interpreting the Enterprise-Specific Configuration Management MIB

- Configuration Management MIB on page 369
- Configuration Management MIB Textual Conventions on page 369
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 370
- Rescue Configuration Change Management Objects on page 372
- Configuration Management Notifications on page 373

Configuration Management MIB

The Juniper Networks enterprise-specific Configuration Management MIB, whose object identifier is **{jnxMIbs 18}**, defines the objects that are used for managing the configuration of Juniper Networks products.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-cfgmgmt.txt .

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

- Configuration Management MIB Textual Conventions on page 369
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 370
- Rescue Configuration Change Management Objects on page 372
- Configuration Management Notifications on page 373

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Configuration Management MIB Textual Conventions

Table 76 on page 370 contains information about the textual conventions used in the enterprise-specific Configuration Management MIB.

Table 76: Configuration Management MIB Textual Conventions

Object	Description	Syntax
jnxCmCfChgSource	Identifies the source of the configuration event.	This object uses the following integer values: • **1**–Other • **2**–CLI • **3**—Junos XML protocol • **4**—Synchronize • **5**—SNMP • **6**—Button • **7**—Autoinstall • **8**—Unknown
jnxCmRescueCfgState	Represents the state of the rescue configuration.	This object uses the following integer values: • **1**—Nonexistent • **2**—Updated

Related Documentation

- Configuration Management MIB on page 369
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 370
- Rescue Configuration Change Management Objects on page 372
- Configuration Management Notifications on page 373

Configuration Change Management Objects and jnxCmCfgChgEventTable

The configuration change management objects listed in Table 77 on page 370 along with the objects in the **jnxCmCfgChgEventTable** provide the configuration change history.

Each **jnxCmCfgChg**, whose object identifier is **{jnxCfgMgmt 1}**, contains the objects listed in Table 77 on page 370.

Table 77: Configuration Change Management Objects

Object	Object ID	Description
jnxCmCfgChgLatestIndex	**jnxCmCfgChg 1**	Contains the index used in **jnxCmCfgChgEventTable** to represent the latest configuration change event.
jnxCmCfgChgLatestTime	**jnxCmCfgChg 2**	Shows the value of **sysUpTime** at the time of the last configuration change. However, this object returns **0** if the management subsystem was reset after the last configuration change.
jnxCmCfgChgLatestDate	**jnxCmCfgChg 3**	Shows the date and time when the configuration was last modified.
jnxCmCfgChgLatestSource	**jnxCmCfgChg 4**	Shows the source from which the configuration event was triggered. This object uses **jnxCmCfChgSource** to represent the source of configuration event. For more information about **jnxCmCfChgSource**, see Table 76 on page 370.

Copyright © 2012, Juniper Networks, Inc.

Table 77: Configuration Change Management Objects *(continued)*

Object	Object ID	Description
jnxCmCfgChgLatestUser	**jnxCmCfgChg 5**	Shows the login name of the current user. This object returns a zero-length string if the user name is not available or not applicable.
jnxCmCfgChgMaxEventEntries	**jnxCmCfgChg 6**	Shows the maximum number of entries that **jnxCmCfgChgEventTable** can contain. Allowable range is **0** though **2147483647**. When the number of entries in **jnxCmCfgChgEventTable** exceeds the maximum value set for **jnxCmCfgChgMaxEventEntries**, the latest entry displaces the oldest entry in the table.

For information about the **jnxCmCfgChgEventTable**, see the following section:

- jnxCmCfgChgEventTable on page 371

jnxCmCfgChgEventTable

The **jnxCmCfgChgEventTable**, whose object identifier is **{jnxCmCfgChg 7}**, contains **jnxCmCfgChgEventEntry** that maps to the most recent configuration change events on the router. The **jnxCmCfgChgMaxEventEntries** object discussed in Table 77 on page 370 controls the number of entries stored in **jnxCmCfgChgEventTable**.

Each **jnxCmCfgChgEventEntry**, whose object identifier is **{jnxCmCfgChgEventTable 1}**, contains the objects listed in Table 78 on page 371.

Table 78: jnxCmCfgChgEventTable

Object	Object ID	Description
jnxCmCfgChgEventIndex	**jnxCmCfgChgEventEntry 1**	Uniquely identifies a configuration change event. The SNMP process assigns monotonically increasing values to each event as it occurs. However, when the SNMP process is reset, the index values also are reset.
jnxCmCfgChgEventTime	**jnxCmCfgChgEventEntry 2**	Contains the value of **sysUpTime** when the event occurred.
jnxCmCfgChgEventDate	**jnxCmCfgChgEventEntry 3**	Contains the system date and time when the event occurred.
jnxCmCfgChgEventSource	**jnxCmCfgChgEventEntry 4**	Shows the source from which the configuration event was triggered. This object uses **JnxCmCfChgSource** to represent the source of configuration event. For more information about **JnxCmCfChgSource**, see Table 76 on page 370.
jnxCmCfgChgEventUser	**jnxCmCfgChgEventEntry 5**	Contains the name of the user who was logged in at the time of the event. Returns a zero-length string if the user name is not applicable or not available.
jnxCmCfgChgEventLog	**jnxCmCfgChgEventEntry 6**	Contains the log of the configuration event. Returns a zero-length string if no log is available.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Configuration Management MIB on page 369
- Configuration Management MIB Textual Conventions on page 369
- Rescue Configuration Change Management Objects on page 372
- Configuration Management Notifications on page 373

Rescue Configuration Change Management Objects

The **jnxCmRescueChg**, whose object identifier is **{jnxCfgMgmt 2}**, contains information about changes to rescue configuration.

Table 79 on page 372 lists the objects associated with **jnxCmRescueChg**.

Table 79: Rescue Configuration Change Management Objects

Object	Object ID	Description
jnxCmRescueChgTime	**jnxCmRescueChg 1**	Contains the value of **sysUpTime** when the rescue configuration was last changed. If the management subsystem has been reset since the last configuration change, this object returns **0**.
jnxCmRescueChgDate	**jnxCmRescueChg 2**	Contains the date and time when the rescue configuration was last changed.
jnxCmRescueChgSource	**jnxCmRescueChg 3**	Shows the source from which the rescue configuration event was triggered. This object uses **jnxCmCfChgSource** to represent the source of configuration event. For more information about **jnxCmCfChgSource**, see Table 76 on page 370.
jnxCmRescueChgUser	**jnxCmRescueChg 4**	Contains the name of the user who was logged in at the time of the event. Returns a zero-length string if the user name is not applicable or not available.
jnxCmRescueChgState	**jnxCmRescueChg 5**	Shows the current state of the rescue configuration. For more information about the different states of rescue configuration, see Table 76 on page 370.

Related Documentation

- Configuration Management MIB on page 369
- Configuration Management MIB Textual Conventions on page 369
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 370
- Configuration Management Notifications on page 373

Copyright © 2012, Juniper Networks, Inc.

Configuration Management Notifications

The Junos OS generates the following traps when a configuration or a rescue configuration event occurs:

- **jnxCmCfgChange**, whose object identifier is **{jnxCmNotificationsPrefix 1}**, contains **jnxCmCfgChgEventTime**, **jnxCmCfgChgEventDate**, **jnxCmCfgChgEventSource**, **jnxCmCfgChgEventUser**, and **jnxCmCfgChgEventLog**.

  NOTE: Because configuration rollback is handled by the master management process that uses the **root** user ID, the **jnxCmCfgChgEventUser** object in the **jnxCmCfgChange** trap always returns **root** as the user name for configuration rollback events.

- **jnxCmRescueChange**, whose object identifier is **{jnxCmNotificationsPrefix 2}**, contains **jnxCmRescueChgTime**, **jnxCmRescueChgDate**, **jnxCmRescueChgSource**, **jnxCmRescueChgUser**, and **jnxCmRescueChgState**.

Related Documentation

- Configuration Management MIB on page 369
- Configuration Management MIB Textual Conventions on page 369
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 370
- Rescue Configuration Change Management Objects on page 372

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 23

Interpreting the Enterprise-Specific Destination Class Usage MIB

- Destination Class Usage MIB on page 375
- jnxDCUsTable on page 376
- jnxDcuStatsTable on page 376

Destination Class Usage MIB

The Juniper Networks enterprise-specific Destination Class Usage (DCU) MIB counts packets from customers by performing a lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

The DCU MIB is a subbranch of the **jnxMibs** branch of the enterprise-specific MIB **{enterprise 2636}** and has an object identifier of **{jnxMIB 6}**. The DCU MIB has one branch, **jnxDCUs**, which contains two tables: **jnxDCUsTable** and **jnxDcuStatsTable**. For information about configuring source and destination class usage, see the *Junos OS Policy Framework Configuration Guide* and *Junos OS Network Interfaces Configuration Guide* . For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-dcu.txt .

NOTE: **Class-based filter match conditions are not supported on J Series Services Routers.**

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

- jnxDCUsTable on page 376
- jnxDcuStatsTable on page 376

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17
- Juniper Networks Enterprise-Specific SNMP Traps on page 125
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113

Copyright © 2012, Juniper Networks, Inc.

jnxDCUsTable

The entries in the **jnxDCUsTable**, whose object identifier is **{jnxDCUTable 1}**, are represented by **jnxDCUsEntry** and are listed in Table 80 on page 376.

Table 80: jnxDCUsEntry

Object	Object Identifier	Description
jnxDCUSrcIfIndex	**jnxDUCsEntry 1**	The interface index of the ingress interface.
jnxDCUDstClassName	**jnxDCUsEntry 2**	The destination class name specified in a routing policy and applied to the forwarding table.
jnxDCUPackets	**jnxDCUsEntry 3**	The number of packets passing through the network.
jnxDCUBytes	**jnxDCUsEntry 4**	The number of bytes passing through the network.

Related Documentation

- Destination Class Usage MIB on page 375
- jnxDcuStatsTable on page 376

jnxDcuStatsTable

jnxDcuStatsTable contains statistics for traffic that satisfies the rules in each configured destination class. A separate set of statistics is kept for each destination class on each interface and address family on which this feature is enabled. This is essentially a replacement for **jnxDCUsTable**.

The entries in the **jnxDcuStatsTable**, whose object identifier is **{jnxDCUs 2}**, are represented by **jnxDCUsStatsEntry** and are listed in Table 81 on page 376.

Table 81: jnxDCUsStatsEntry

Ojbect	Object Identifier	Description
jnxDcuStatsSrcIfIndex	**jnxDcuStatsEntry** 1	The interface index of the ingress interface for traffic counted in each entry.
jnxDcuStatsAddrFamily	**jnxDcuStatsEntry 2**	The address family of the entry's traffic.
nxDcuStatsClassName	**jnxDcuStatsEntry 3**	The name of the destination class that applies to the entry's traffic.
jnxDcuStatsPackets	**jnxDcuStatsEntry 4**	The number of packets received on this interface and belonging to this address family that match this destination class.

Copyright © 2012, Juniper Networks, Inc.

Table 81: jnxDCUsStatsEntry *(continued)*

Ojbect	Object Identifier	Description
jnxDcuStatsBytes	**jnxDcuStatsEntry 5**	The number of bytes received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsClName	**jnxDcuStatsEntry 6**	The name of the destination class. This object is a duplicate of **jnxDcuStatsClassName** and is included to satisfy those network management applications that cannot extract the destination class name from the instance portion of the OID.

Related Documentation

- Destination Class Usage MIB on page 375
- jnxDCUsTable on page 376

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 24

Interpreting the Enterprise-Specific DNS Objects MIB

- DNS Objects MIB on page 379
- jnxJsDnsProxyDataObjects on page 379

DNS Objects MIB

The Juniper Networks enterprise-specific DNS Objects MIB, **jnxJsDns**,whose object ID is **{jnxJsDnsRoot 1}**, provides collated statistics for the Domain Name System (DNS) proxy collected over all interfaces on which it is configured to serve.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-dns.txt .

For information about the enterprise-specific DNS Objects MIB, see the following topic:

- jnxJsDnsProxyDataObjects on page 379

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsDnsProxyDataObjects

jnxJsDnsProxyDataObjects, whose object ID is **{jnxJsDns 1}**, displays the DNS query-related statistics listed in Table 82 on page 379.

Table 82: jnxJsDnsProxyDataObjects

Object	Object ID	Description
jnxJsDNSProxyQueriesReceived	**jnxJsDnsProxyDataObjects 1**	Total number of DNS queries received by the DNS proxy.
jnxJsDnsProxyResponsesSent	**jnxJsDnsProxyDataObjects 2**	Number of DNS queries answered sent by the DNS proxy. This includes DNS cache hits and misses that were answered.

Copyright © 2012, Juniper Networks, Inc.

Table 82: jnxJsDnsProxyDataObjects *(continued)*

Object	Object ID	Description
jnxJsDnsProxyQueriesForwarded	**jnxJsDnsProxyDataObjects 3**	Number of DNS queries forwarded to other DNS servers. This is the number of queries that have been proxied due to cache misses.
jnxJsDnsProxyNegativeResponses	**jnxJsDnsProxyDataObjects 4**	Number of negative DNS query responses. This is the count of DNS queries for which the proxy could not obtain answers.
jnxJsDnsProxyRetryRequests	**jnxJsDnsProxyDataObjects 5**	Number of DNS retry queries that this proxy received.
jnxJsDnsProxyPendingRequests	**jnxJsDnsProxyDataObjects 6**	Number of DNS requests yet to be answered.
jnxJsDnsProxyServerFailures	**jnxJsDnsProxyDataObjects 7**	Number of DNS proxy failures.

Related Documentation

- DNS Objects MIB on page 379

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 25

Interpreting the Enterprise-Specific Ethernet MAC MIB

- Ethernet MAC MIB on page 381
- jnxMacStatsTable on page 381

Ethernet MAC MIB

The Juniper Networks enterprise-specific Ethernet Media Access Control (MAC) MIB, whose object identifier is **{jnxMibs 23}**, monitors media access control statistics on Gigabit Ethernet intelligent queuing (IQ) interfaces. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-mac.txt .

For information about the enterprise-specific Ethernet MAC MIB objects, see the following topic:

- jnxMacStatsTable on page 381

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxMacStatsTable

The **jnxMacStatsTable** contains a list of MAC statistics for Gigabit Ethernet interfaces.

jnxMacStatsEntry objects are listed in Table 83 on page 381.

Table 83: jnxMacStatsTable

Object	Object Identifier	Description
jnxVlanIndex	**jnxMacStatsEntry 1**	The virtual LAN (VLAN) ID of a VLAN.
jnxSourceMacAddress	**jnxMacStatsEntry 2**	The source MAC address.
jnxMacHCInOctets	**jnxMacStatsEntry 3**	The number of total octets received in this VLAN/MAC address.

Table 83: jnxMacStatsTable *(continued)*

Object	Object Identifier	Description
jnxMacHCInFrames	**jnxMacStatsEntry 4**	The number of total frames received in this VLAN/MAC address
jnxMacHCOutOctets	**jnxMacStatsEntry 5**	The number of total octets transmitted in this VLAN/MAC address.
jnxMacHCOutFrames	**jnxMacStatsEntry 6**	The number of total frames transmitted in this VLAN/MAC address.

Related Documentation

- Ethernet MAC MIB on page 381

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 26

Interpreting the Enterprise-Specific Event MIB

- Event MIB on page 383
- jnxEventAvTable on page 383
- Notifications for the Event MIB on page 384

Event MIB

The Juniper Networks enterprise-specific Event MIB, whose object identifier is **{jnxMibs 37}**, defines a generic trap that can be generated using an op script or event policy. This MIB provides the ability to specify a system log string and raise a trap if that system log string is found.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-event.txt .

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

- jnxEventAvTable on page 383
- Notifications for the Event MIB on page 384

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxEventAvTable

The **jnxEventAvTable**, whose object identifier is **{jnxEventNotifyVars 2}**, provides information about traps generated by op scripts or event policies.

jnxEventAvEntry, whose object identifier is **{jnxEventAvTable 1}**, has three objects, which are listed in Table 84 on page 384.

Table 84: jnxEventAvTable

Object	Object Identifier	Description
jnxEventAvIndex	**jnxEventAvEntry 1**	The sequence number of the attribute value pair in the trap generated by a op script or event policy.
jnxEventAvAttribute	**jnxEventAvEntry 2**	The attribute name in the trap generated by an op script or event policy.
jnxEventAvValue	**jnxEventAvEntry 3**	The value of the attribute identified by **jnxEventAvAttribute**.

Related Documentation

- Event MIB on page 383
- Notifications for the Event MIB on page 384

Notifications for the Event MIB

Table 85 on page 384 lists the supported notifications for the Event MIB.

Table 85: Supported Notifications for the Event MIB

Object	Object Identifier	Description
jnxEventTrapDescr	**jnxEventNotificationPrefix 1**	A notification generated by an op script or event policy. In addition to the **jnxEventTrap** objects, this notification can include one or more attribute value pairs (identified by **jnxEventAvAttribute** and **jnxEventAvValue**).

Related Documentation

- Event MIB on page 383
- jnxEventAvTable on page 383

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 27

Interpreting the Enterprise-Specific Firewall MIB

- Firewall MIB on page 385
- jnxFirewallsTable on page 386
- jnxFirewallCounterTable on page 386

Firewall MIB

The Juniper Networks enterprise-specific Firewall MIB, whose object identifier is **{jnxMibs 5}**, contains information about firewall filters and policies.

Firewall MIB contains two tables, **jnxFirewallsTable** and **jnxFirewallCounterTable**.

The **jnxFirewallsTable** does not support the following conditions:

- Counter and filter names that have more than 24 characters.
- Duplicate counter names, even if the counter types are different.

Because of the preceding limitations, the **jnxFirewallsTable** has been deprecated and replaced with **jnxFirewallCounterTable**. However, for backward compatibility, the **jnxFirewallsTable** is retained in the Firewall MIB.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-firewall.txt .

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

- jnxFirewallsTable on page 386
- jnxFirewallCounterTable on page 386

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxFirewallsTable

The deprecated **jnxFirewallsTable** contains **jnxFirewallsEntry**, whose object ID is **{jnxFirewallsTable 1}**. Each **jnxFirewallsEntry** contains the objects listed in Table 86 on page 386.

Table 86: jnxFirewallsEntry

Object	Object ID	Description
jnxFWFilter	**jnxFirewallsEntry 1**	The name of the firewall filter. This object does not support names that have more than 24 characters.
jnxFWCounter	**jnxFirewallsEntry 2**	The name of the counter or the policer. This name is specific within the firewall filter.
jnxFWType	**jnxFirewallsEntry 3**	The type of the **jnxFWCounter** object. The value of **jnxFWType** can be **1** (other), **2** (counter), or **3** (policer).
jnxFWPackets	**jnxFirewallsEntry 4**	The number of packets that are associated with the specified counter or policer.
jnxFWBytes	**jnxFirewallsEntry 5**	The number of bytes that are associated with the counter. For policers, the value of **jnxFWBytes** is always zero because the policers do not count the number of bytes.

Related Documentation

- Firewall MIB on page 385
- jnxFirewallCounterTable on page 386

jnxFirewallCounterTable

The **jnxFirewallCounterTable**, whose object identifier is **{jnxFirewalls 2}**, replaces the deprecated **jnxFirewallsTable**. Each **jnxFirewallCounterEntry** contains the objects listed in Table 87 on page 386.

Table 87: jnxFirewallCounterEntry

Object	Object ID	Description
jnxFWCounterFilterName	**jnxFirewallCounterEntry 1**	The name of the firewall filter. The name can have up to 127 characters.
jnxFWCounterName	**jnxFirewallCounterEntry 2**	The name of the counter or the policer. The name can have up to 127 characters.
jnxFWCounterType	**jnxFirewallCounterEntry 3**	The type of the **jnxFWCounterName** object. The value of **jnxFWType** can be **1** (other), **2** (counter), or **3** (policer).

Copyright © 2012, Juniper Networks, Inc.

Table 87: jnxFirewallCounterEntry *(continued)*

Object	Object ID	Description
jnxFWCounterPacketCount	jnxFirewallCounterEntry 4	The number of packets that are associated with the specified counter or policer.
jnxFWCounterByteCount	jnxFirewallCounterEntry 5	The number of bytes that are associated with the counter or policer.
jnxFWCounterDisplayFilterName	jnxFirewallCounterEntry 6	The name of the firewall filter. The name can have up to 127 characters.
jnxFWCounterDisplayName	jnxFirewallCounterEntry 7	The name of the counter or the policer.
jnxFWCounterDisplayType	jnxFirewallCounterEntry 8	The type of the **jnxFWCounterName** object. The value of **jnxFWType** can be **1** (other), **2** (counter), or **3** (policer).

Related Documentation

- Firewall MIB on page 385
- jnxFirewallsTable on page 386

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 28

Interpreting the Enterprise-Specific Host Resources MIB

- Host Resources MIB on page 389
- jnxHrStorageTable on page 389

Host Resources MIB

The Juniper Networks enterprise-specific Host Resources MIB, whose object identifier is **{jnxMibs 31}**, extends the **hrStorageTable** defined in RFC 2790, the standard Host Resources MIB, to include the **jnxHrStoragePercentUsed** object.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-hostresources.txt .

For information about the enterprise-specific Host Resources MIB objects, see the following topic:

- jnxHrStorageTable on page 389

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxHrStorageTable

The **jnxHrStorageTable**, whose object identifier is **{jnxHrStorage 1}**, contains **jnxHrStorageEntry**. Each **jnxHrStorageEntry** augments the **hrStorageEntry** to provide additional file system data, and contains the following object:

- **jnxHrStoragePercentUsed**—Object identifier is **{jnxHrStorageEntry 1}**. —Shows what percentage of the total storage space has been used.

Related Documentation

- Host Resources MIB on page 389

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 29

Interpreting the Enterprise-Specific Interface MIB

- Interface MIB on page 391
- jnxIfTable on page 391
- ifChassisTable on page 394

Interface MIB

The Juniper Networks enterprise-specific Interface MIB extends the standard **ifTable** (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-if-extensions.txt .

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

- jnxIfTable on page 391
- ifChassisTable on page 394

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxIfTable

jnxIfTable lists traffic statistics, input errors, and output errors for an interface.

jnxIfEntry objects are listed in Table 88 on page 391.

Table 88: jnxIfTable

Object	Object Identifier	Description
ifIn1SecRates	**ifJnxEntry 1**	The number of bits per second (bps) delivered by this sublayer to its next higher sublayer.

Table 88: jnxIfTable *(continued)*

Object	Object Identifier	Description
ifIn1SecOctets	**ifJnxEntry 2**	The number of octets per second delivered by this sublayer to its next higher sublayer.
ifIn1SecPkts	**ifJnxEntry 3**	The number of packets per second (pps) delivered by this sublayer to its next higher sublayer.
ifOut1SecRate	**ifJnxEntry 4**	The number of bps delivered by this sublayer to its next lower sublayer.
ifOut1SecOctets	**ifJnxEntry 5**	The number of octets per second delivered by this sublayer to its next lower sublayer.
ifOut1SecPkts	**ifJnxEntry 6**	The number of pps delivered by this sublayer to its next lower sublayer.
ifHCIn1SecRate	**ifJnxEntry 7**	The number of bps delivered by this sublayer to its next higher sublayer. This object is a 64-bit version of **ifIn1SecRate**.
ifHCOut1SecRate	**ifJnxEntry 8**	The number of bps delivered by this sublayer to its next lower sublayer. This object is a 64-bit version of **ifOut1SecRate**.
ifJnxInErrors	**ifJnxEntry 9**	Errors: The sum of the incoming frame aborts and FCS errors.
ifJnxInFrameErrors	**ifJnxEntry 10**	Framing Errors: The number of input packets that were misaligned.
ifJnxInQDrops	**ifJnxEntry 11**	Drops: The number of packets dropped by the input queue of the I/O Manager ASIC.
ifJnxInRunts	**ifJnxEntry 12**	Runts: Frames received that are smaller than the runt threshold.
ifJnxInGiants	**ifJnxEntry 13**	Giants: Frames received that are larger than the giant threshold.
ifJnxInDiscards	**ifJnxEntry 14**	Policed discards: Frames that the incoming packet match code discarded because they were not recognized or of interest.
ifJnxInHslCrcErrors	**ifJnxEntry 15**	HS link CRC errors: The number of CRC errors on the high-speed links between the ASICs responsible for handling the router interfaces while receiving packets.
ifJnxInHslFifoOverFlows	**ifJnxEntry 16**	HS link FIFO overflows: The number of FIFO overflows on the high-speed links between the ASICs responsible for handling the router interfaces.

Copyright © 2012, Juniper Networks, Inc.

Table 88: jnxIfTable *(continued)*

Object	Object Identifier	Description
ifJnxInL3Incompletes	**ifJnxEntry 17**	L3 incompletes: The number of incoming packets that fail Layer 3 sanity checks of the header.
ifJnxInL2ChanErrors	**ifJnxEntry 18**	L2 channel errors: The number of incoming packets for which the software could not find a valid logical interface.
ifJnxInL2MismatchTimeouts	**ifJnxEntry 19**	L2 mismatch timeouts: The count of malformed or short packets that cause the incoming packet handler to discard the frame as unreadable.
ifJnxInInvalidVCs	**ifJnxEntry 20**	Invalid virtual circuits (VCs): The number of cells that arrived for a nonexistent VC.
ifJnxInFifoErrors	**ifJnxEntry 21**	FIFO errors: The number of FIFO errors in the received direction as reported by the ASIC on the PIC.
ifJnxBucketDrops	**ifJnxEntry 22**	Bucket drops: Drops because traffic load exceeded the interface transmit and receive leaky bucket configuration.
ifJnxSramErrors	**ifJnxEntry 23**	SRAM errors: This counter increments when a hardware error has occurred in the SRAM on the PIC.
ifJnxOutErrors	**ifJnxEntry 24**	Errors: The sum of the outgoing frame aborts and FCS errors.
ifJnxCollisions	**ifJnxEntry 25**	Collisions: The number of output collisions detected on this interface.
ifJnxCarrierTrans	**ifJnxEntry 26**	Carrier transitions: The number of times the interface saw the carrier signal transition.
ifJnxOutQDrops	**ifJnxEntry 27**	Drops: The number of packets dropped by the output queue of the I/O Manager ASIC.
ifJnxOutAgedErrors	**ifJnxEntry 28**	Aged packets: The number of packets that remained in shared packet SDRAM for so long that the system automatically purged them.
ifJnxOutFifoErrors	**ifJnxEntry 29**	FIFO errors: The number of FIFO errors in the transmit direction as reported by the ASIC on the PIC.
ifJnxOutHslFifoUnderFlows	**ifJnxEntry 30**	HS link FIFO underflows: The number of FIFO underflows on the high-speed links between the ASICs responsible for handling the router interfaces.

Copyright © 2012, Juniper Networks, Inc.

Table 88: jnxIfTable *(continued)*

Object	Object Identifier	Description
ifJnxOutHslCrcErrors	**ifJnxEntry 31**	HS link CRC errors: The number of CRC errors on the high-speed links between the ASICs responsible for handling the router interfaces while transmitting packets.
ifJnxCrcErrors	**ifJnxEntry 32**	The count of the number of CRC errors that have occurred.
ifJnxFcsErrors	**ifJnxEntry 33**	The count of the number of FCS errors that have occurred.

Related Documentation

- Interface MIB on page 391
- ifChassisTable on page 394

ifChassisTable

ifChassisTable provides additional interface and chassis information.

ifChassisEntry objects are listed in Table 89 on page 394.

Table 89: ifChassisTable

Object	Object Identifier	Description
ifChassisFpc	**ifChassisEntry 1**	The number of the FPC card on which the interface is located in the chassis. It is the chassis slot in which the FPC card is installed for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical FPC associated with the specified interface.
ifChassisPic	**ifChassisEntry 2**	The number of the PIC card on which the interface is located in the chassis. It is the PIC location on the FPC card for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical PIC associated with the specified interface.

Copyright © 2012, Juniper Networks, Inc.

Table 89: ifChassisTable *(continued)*

Object	Object Identifier	Description
ifChassisPort	**ifChassisEntry 3**	The number of the port on the PIC card on which the interface is located in the chassis. It is the port number on the PIC card for the specified interface. Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical port associated with the specified interface.
ifChassisChannel	**ifChassisEntry 4**	The channel identifier for the specified interface if it is part of a channelized interface. Although the channel is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to network management convention. For an interface that could not be channelized, this object returns zero.
ifChassisLogicalUnit	**ifChassisEntry 5**	The logical unit number of the specified interface. It is the logical part of the interface that is configured on the physical or channel part, if any. Although the logical unit number is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to network management convention. For an interface that is really a physical device, this value returns zero.
ifChassisPicIndex	**ifChassisEntry 6**	The indexes for the Chassis MIB tables. This is the instance index that keys into **jnxContentsTable** in the Chassis MIB. For example, the octet string of 8.1.2.0 means a PIC (“ 8&” first digit) at FPC slot 0 (“1–1” , second digit minus one if nonzero) PIC number 1 (“2–1” , third digit) minus one if nonzero port number, whatever (fourth digit currently unused). In turn, this PIC index can be plugged in by the NMS directly after any MIB objects in the **jnxContentsTable** obtain that PIC object for the specified interface. This object is valid only for interfaces having real and physical PIC cards. Otherwise, it returns an octet string “0.0.0.0.”

Related Documentation

- Interface MIB on page 391
- jnxIfTable on page 391

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 30

Interpreting the Enterprise-Specific IP Forward MIB

- IP Forward MIB on page 397
- jnxIpCidrRouteTable on page 398

IP Forward MIB

The Juniper Networks enterprise-specific IP Forward MIB, whose object identifier is **{jnxMibs 38}**, extends the **ipCidrRouteTable** in the IP Forwarding Table MIB (as defined in RFC 2096) to include a tunnel name when the next hop is through an RSVP-signaled LSP.

This MIB adds an **jnxIpCidrRouteTunnelName** attribute to the **ipCidrRouteTable**. The attribute exists for each entry in the **ipCidrRouteTable**. (One entry in the **ipCidrRouteTable** represents each route in **inet.0**). If the route's next hop is an RSVP-signaled MPLS LSP, the new attribute contains the LSP name. If the route's next hop is not an RSVP-signaled MPLS LSP, the new attribute is defined as null.

The attribute's name is **jnxIpCidrRouteTunnelName**. Its object identifier (OID) is 1.3.6.1.4.1.2636.3.38.1.1.1. As with any SNMP attribute, an index is appended to the OID to form the instance identifier. Because this attribute augments the **ipCidrRouteTable**, the index is identical to that used in the **ipCidrRouteTable**. The index is formed by concatenating the destination address, subnet mask, ToS byte, and next hop.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-ipforward.txt .

For more information about the enterprise-specific IP Forward MIB objects, see the following topic:

- jnxIpCidrRouteTable on page 398

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxIpCidrRouteTable

The **jnxIpCidrRouteTable**, whose object identifier is **{jnxIpForwardMIB 1}**, extends the **ipCidrRouteTable** with additional data.

jnxIpCidrRouteEntry, whose object identifier is **{jnxIpCidrRouteTable 1}**, has one object, which is listed in Table 90 on page 398.

Table 90: jnxIpCidrRouteTable

Object	Object Identifier	Description
jnxIpCidrRouteTunnelName	**nxIpCidrRouteEntry 1**	The canonical name assigned to the tunnel. The router forwards traffic bound for the destination through this tunnel.

Related Documentation

- IP Forward MIB on page 397

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 31

Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object MIB

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

IPsec Generic Flow Monitoring Object MIB

The Juniper Networks enterprise-specific IPsec Generic Flow Monitoring Object MIB, whose object ID is **{jnxIpSecMibRoot 1}**, defines the objects used to monitor the entries pertaining to IPsec objects and the management of the IPsec VPN functionalities. This generic MIB models the standard, dynamic aspects of IPsec, including the counters and objects that are of management interest in a standard IPsec implementation.

This MIB module is based on the **jnxIpSecMonitorMib**. Building on the existing Internet Key Exchange (IKE) infrastructure, the security IKE implementation integrates the value-added features for the security products.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt .

For information about the enterprise-specific IPsec Generic Flow Monitoring Object MIB, see the following topics:

- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404

Copyright © 2012, Juniper Networks, Inc.

- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Branch Tree Objects

The following branch tree objects are associated with the IPsec Generic Flow Monitoring Object MIB:

- **jnxIpSecFlowMonNotifications {jnxIpSecFlowMonMIB 0}**
- **jnxIpSecFlowMonPhaseOne {jnxIpSecFlowMonMIB 1}**
- **jnxIpSecFlowMonPhaseTwo {jnxIpSecFlowMonMIB 2}**

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

IPSec Generic Flow Monitoring Object MIB Textual Conventions

- **JnxIkePeerType**—The type of IPsec Phase 1 IKE peer identity. This identity is the local IKE identity to send in the exchange. The IKE peer can be identified by one of the ID types defined in IPsec DOI (see Table 91 on page 400).

Table 91: IKE Identity Type Textual Conventions

Type	Description	Syntax Integer
Unknown	Unknown type	**(0)**
idIpv4Addr	IPv4 address	**(1)**
idFqdn	Fully qualified domain name	**(2)**
idDn	Distinguished name	**(3)**
idUfqdn	User fully qualified domain name	**(4)**

- **JnxIkeNegoMode**—The IPsec Phase 1 IKE negotiation mode (see Table 92 on page 401).

Copyright © 2012, Juniper Networks, Inc.

Table 92: IKE Negotiation Mode Text Conventions

Type	Description	Syntax Integer
Main mode	A six-message Phase 1 exchange that provides identity protection	(1)
Aggressive mode	A three-message Phase 1 exchange that does not provide identity protection	(2)

- **JnxIkeHashAlgo**—The hash algorithm used in IPsec Phase 1 IKE negotiations (see Table 93 on page 401).

Table 93: IKE Negotiations Hash Alogorithms

Hash Alogorithm	Syntax Integer
md5	(1)
sha	(2)

- **JnxIkeAuthMethod**—The authentication method used in IPsec Phase 1 IKE negotiations (see Table 94 on page 401).

Table 94: IKE Authentication Method

Method	Syntax Integer
preSharedKey	(1)
dssSignature	(2)
rsaSignature	(3)
rsaEncryption	(4)
revRsaEncryption	(5)
xauthPreSharedKey	(6)
xauthDssSignature	(7)
xauthRsaSignature	(8)
xauthRsaEncryption	(9)
xauthRevRsaEncryption	(10)

Copyright © 2012, Juniper Networks, Inc.

- **JnxIkePeerRole**—The role of the local endpoint in negotiating the IPsec Phase 1 IKE security association (SA). It can be either initiator or responder (see Table 95 on page 402).

Table 95: Role of Local Endpoint in Negotiations

Role	Syntax Integer
Initiator	(1)
Responder	(2)

- **JnxIkeTunStateType**—State of the Phase 1 IKE negotiation (see Table 96 on page 402).

Table 96: State of Phase 1 IKE Negotiation

State	Syntax Integer
Up	(1)
Down	(2)

- **JnxDiffHellmanGrp**—The Diffie-Hellman Group used in negotiations (see Table 97 on page 402).

Table 97: Diffie-Hellman Group in Negotiations

Diffie Hellman Group	Description	Syntax Integer
Unknown	Unknown	(0)
modp768	768-bit MODP	(1)
modp1024	1024-bit MODP	(2)
modp1536	modp1536	(3)

- **JnxKeyType**—The type of key used by an IPsec Phase 2 Tunnel (see Table 98 on page 402).

Table 98: Key Used by IPsec Phase 2 Tunnel

Key	Syntax Integer
Unknown	(0)
keyIke	(1)
keyManual	(2)

- **JnxEncryptAlgo** —The encryption algorithm used in negotiations (see Table 99 on page 403).

Copyright © 2012, Juniper Networks, Inc.

Table 99: Encryption Algorithm in Negotiations

Algorithm	Syntax Integer
espDes	**(1)**
esp3des	**(2)**
espNull	**(3)**
espAes128	**(4)**
espAes192	**(5)**
espAes256	**(6)**

- **JnxAuthAlgo**—The authentication algorithm used by an SA of an IPsec Phase 2 Tunnel (see Table 100 on page 403).

Table 100: Role of Local Endpoint in Negotiations

Algorithm	Syntax Integer
Unknown	**(0)**
hmacMd5	**(1)**
hmacSha	**(2)**

- **JnxRemotePeerType**—The type of the remote peer gateway (endpoint) (see Table 101 on page 403).

Table 101: Type of Remote Peer Gateway

Gateway Type	Syntax Integer and Description
Unknown	**(0)**
Static	**(1)** Static (remote peer whose IP address is known beforehand)
Dynamic	**(2)** Dynamic (remote peer whose IP address is not known beforehand)

- **JnxSpiType**—The type of the SPI associated with IPsec Phase 2 SAs. An unsigned 32-bit integer (256. . . 4294967295).
- **JnxSAType**—The SA type (see Table 102 on page 404).

Table 102: Role of Local Endpoint in Negotiations

SA Type	Syntax Integer
Unknown	(0)
Manual	(1)
Dynamic	(2)

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

Number of IKE Tunnels Currently Active

NOTE: **The jnxIkeNumOfTunnels object is not supported in this release.**

Table 103 on page 404 reports the number of IKE Tunnels currently active.

Table 103: Number of IKE Tunnels Currently Active

Object	Object ID	Description
jnxIkeNumOfTunnels	**jnxIpSecFlowMonPhaseOne 1**	Number of IKE Tunnels (Phase 1) actively negotiating between peers. The SA can be in either the up or down state. This attribute details the number of IKE tunnels in **jnxIkeTunnelMonTable**.

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

IPsec Phase 1 IKE Tunnel Table

Table 104 on page 405 identifies objects listed in the IPsec Phase 1 IKE Tunnel Table.

Copyright © 2012, Juniper Networks, Inc.

Phase 1 is used to negotiate the parameter and key material required to establish an ISAKMP SA.

Phase 1 SA components include an encryption algorithm, authentication, Diffie-Hellman group values, and anthentication methods, such as preshared keys or certificates.

Table 104: IPsec Phase 1 IKE Tunnel Table

Object	Object ID	Description
jnxIkeTunnelMonTable	**jnxIpSecFlowMonPhaseOne 2**	The IPsec Phase 1 IKE Tunnel Table. There is one entry in this table for each active IPsec Phase 1 IKE tunnel.
jnxIkeTunnelMonEntry	**jnxIkeTunnelMonTable 1**	Attributes associated with an active IPsec Phase 1 IKE tunnel. Sequence of attributes: • **jnxIkeTunMonRemoteGwAddrType** • **jnxIkeTunMonRemoteGwAddr** • **jnxIkeTunMonIndex** • **jnxIkeTunMonLocalGwAddrType** • **jnxIkeTunMonLocalGwAddr** • **jnxIkeTunMonState** • **jnxIkeTunMonInitiatorCookie** • **jnxIkeTunMonResponderCookie** • **jnxIkeTunMonLocalRole** • **jnxIkeTunMonLocalIdType** • **jnxIkeTunMonLocalIdValue** • **jnxIkeTunMonLocalCertName** • **jnxIkeTunMonRemoteIdType** • **jnxIkeTunMonRemoteIdValue** • **jnxIkeTunMonNegoMode** • **jnxIkeTunMonDiffHellmanGrp** (not supported in this release) • **jnxIkeTunMonEncryptAlgo** • **jnxIkeTunMonHashAlgo** • **jnxIkeTunMonAuthMethod** • **jnxIkeTunMonLifeTime** • **jnxIkeTunMonActiveTime** • **jnxIkeTunMonInOctets** • **jnxIkeTunMonInPkts** • **jnxIkeTunMonOutOctets** • **jnxIkeTunMonOutPkts** • **jnxIkeTunMonXAuthUserId** • **jnxIkeTunMonDPDDownCount**
jnxIkeTunMonRemoteGwAddrType	**jnxIkeTunnelMonEntry 1**	IP address type of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel.

Table 104: IPsec Phase 1 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxJsFwAuthClientIpAddr	jnxJsAuthTrapVars 4	IP address of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel.
jnxIkeTunMonIndex	jnxIkeTunnelMonEntry 3	Index number of IPsec Phase 1 IKE Tunnel Table. The index number begins at 1 and is incremented with each tunnel that is created. The value of this object will wrap at 2,147,483,647.
jnxIkeTunMonLocalGwAddr	jnxIkeTunnelMonEntry 4	IP address of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel.
jnxIkeTunMonLocalGwAddrType	jnxIkeTunnelMonEntry 5	IP address type of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel.
jnxIkeTunMonState	jnxIkeTunnelMonEntry 6	State of IKE tunnel. It can be: • 1—up, negotiation completed. • 2—down, being negotiated.
jnxIkeTunMonInitiatorCookie	jnxIkeTunnelMonEntry 7	Cookie as generated by peer that initiated the IKE Phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunMonResponderCookie	jnxIkeTunnelMonEntry 8	Cookie as generated by peer responding to the IKE Phase 1 negotiation initiated by the remote peer. This cookie is carried in the ISAKMP header.
jnxIkeTunMonLocalRole	jnxIkeTunnelMonEntry 9	Role of local peer identity. The role of the local peer can be: • Initiator • Responder
jnxIkeTunMonLocalIdType	jnxIkeTunnelMonEntry 10	Type of local peer identity. The local peer can be identified by: • IP address • Fully qualified domain name string • Distinguished name string
jnxIkeTunMonLocalIdValue	jnxIkeTunnelMonEntry 11	Value of local peer identity. If the local peer type is an IP address, then this is the IP address used to identify the local peer. If the local peer type is a fully qualified domain name string, then this is the fully qualified domain name string of the local peer. If the local peer type is a distinguished name string, then this is the distinguished name string of the local peer.

Copyright © 2012, Juniper Networks, Inc.

Table 104: IPsec Phase 1 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxIkeTunMonLocalCertName	**jnxIkeTunnelMonEntry 12**	Name of certificate used for authentication of the local tunnel endpoint. This object has some valid value only if the negotiated IKE authentication method is other than preshared key. If the IKE negotiation does not use a certificate-based authentication method, then the value of this object is a NULL string.
jnxIkeTunMonRemoteIdType	**jnxIkeTunnelMonEntry 13**	Type of remote peer identity. The remote peer can be identified by: • IP address • Fully qualified domain name string • Distinguished name string
jnxIkeTunMonRemoteIdValue	**jnxIkeTunnelMonEntry 14**	Value of remote peer identity. If the remote peer type is an IP address, then this is the IP address used to identify the remote peer. If the remote peer type is a fully qualified domain name string, then this is the fully qualified domain name string of the remote peer. If the remote peer type is a distinguished name string, then this is the distinguished name string of the remote peer.
jnxIkeTunMonNegoMode	**jnxIkeTunnelMonEntry 15**	Negotiation mode of IPsec Phase 1 IKE tunnel.
NOTE: The **jnxIkeTunMonDiffHellmanGrp** object is not supported in this release.		
jnxIkeTunMonDiffHellmanGrp	**jnxIkeTunnelMonEntry 16**	Diffie-Hellman Group used in IPsec Phase 1 IKE negotiations.
jnxIkeTunMonEncryptAlgo	**jnxIkeTunnelMonEntry 17**	Encryption algorithm used in IPsec Phase 1 IKE negotiations.
jnxIkeTunMonHashAlgo	**jnxIkeTunnelMonEntry 18**	Hash algorithm used in IPsec Phase 1 IKE negotiations.
jnxIkeTunMonAuthMethod	**jnxIkeTunnelMonEntry 19**	Authentication method used in IPsec Phase 1 IKE negotiations.
jnxIkeTunMonLifeTime	**jnxIkeTunnelMonEntry 20**	Negotiated lifetime of IPsec Phase 1 IKE tunnel in seconds.
jnxIkeTunMonActiveTime	**jnxIkeTunnelMonEntry 21**	Length of time IPsec Phase 1 IKE tunnel has been active in hundredths of seconds.
jnxIkeTunMonInOctets	**jnxIkeTunnelMonEntry 22**	Total number of octets received by this IPsec Phase 1 IKE SA.
jnxIkeTunMonInPkts	**jnxIkeTunnelMonEntry 23**	Total number of packets received by this IPsec Phase 1 IKE SA.

Copyright © 2012, Juniper Networks, Inc.

Table 104: IPsec Phase 1 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxIkeTunMonOutOctets	**jnxIkeTunnelMonEntry 24**	Total number of octets sent by this IPsec Phase 1 IKE SA.
jnxIkeTunMonOutPkts	**jnxIkeTunnelMonEntry 25**	Total number of packets sent by this IPsec Phase 1 IKE SA.
jnxIkeTunMonXAuthUserId	**jnxIkeTunnelMonEntry 26**	Extended Authentication (XAuth) User Identifier. Identifies the user associated with this IPsec Phase 1 negotiation.
jnxIkeTunMonDPDDownCount	**jnxIkeTunnelMonEntry 27**	Number of times that the remote peer is detected in a dead (or down) state.

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408
- IPsec Phase 2 Security Association Table on page 411

IPsec Phase 2 IKE Tunnel Table

Table 105 on page 408 identifies objects listed in the IPsec Phase 2 IKE Tunnel Table.

During this phase, IKE negotiates IPsec SA parameters and setup, matching IPsec SA in the peers.

Phase 2 VPN includes tunnel peer connection, associated with a specific policy or a tunnel interface. Phase 2 SA components include encryption and authentication algorithms, proxy-IDs, and optional DH group values.

Table 105: IPsec Phase 2 IKE Tunnel Table

Object	Object ID	Description
NOTE: The **jnxIpSecNumOfTunnels** object is not supported in this release.		
jnxIpSecNumOfTunnels	**jnxIpSecFlowMonPhaseTwo 1**	Number of IPsec VPN tunnels. This attribute should report the number of IPsec VPN tunnels in **jnxIpSecTunnelTable**.
jnxIpSecTunnelMonTable	**jnxIpSecFlowMonPhaseTwo 2**	The IPsec Phase 2 Tunnel Table. There is one entry in this table for each active IPsec Phase 2 tunnel. If the tunnel is terminated, then the entry is no longer available after the table has been refreshed.

Copyright © 2012, Juniper Networks, Inc.

Table 105: IPsec Phase 2 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxIpSecTunnelMonEntry	**jnxIpSecTunnelMonTable 1**	Each entry contains the attributes associated with an active IPsec Phase 2 tunnel. Sequence of attributes: • **jnxIpSecTunMonRemoteGwAddrType** • **jnxIpSecTunMonRemoteGwAddr** • **jnxIpSecTunMonIndex** • **jnxIpSecTunMonLocalGwAddrType** • **jnxIpSecTunMonLocalGwAddr** • **jnxIpSecTunMonLocalProxyId** • **jnxIpSecTunMonRemoteProxyId** • **jnxIpSecTunMonKeyType** • **jnxIpSecTunMonRemotePeerType** • **jnxIpSecTunMonOutEncryptedBytes** • **jnxIpSecTunMonOutEncryptedPkts** • **jnxIpSecTunMonInDecryptedBytes** • **jnxIpSecTunMonInDecryptedPkts** • **jnxIpSecTunMonAHInBytes** • **jnxIpSecTunMonAHInPkts** • **jnxIpSecTunMonAHOutBytes** • **jnxIpSecTunMonAHOutPkts** • **jnxIpSecTunMonReplayDropPkts** • **jnxIpSecTunMonAhAuthFails** • **jnxIpSecTunMonDecryptFails** • **jnxIpSecTunMonBadHeaders** • **jnxIpSecTunMonBadTrailers** • **jnxIkeTunMonOutOctets** • **jnxIpSecTunMonDroppedPkts** (not supported in this release)
jnxIpSecTunMonRemoteGwAddrType	**jnxIpSecTunnelMonEntry 1**	IP address type of remote gateway (endpoint) for the IPsec Phase 2 tunnel.
jnxIpSecTunMonRemoteGwAddr	**jnxIpSecTunnelMonEntry 2**	IP address of remote gateway (endpoint) for the IPsec Phase 2 tunnel.
jnxIpSecTunMonIndex	**jnxIpSecTunnelMonEntry 3**	Index number of IPsec Phase 2 Tunnel Table. The index number begins at 1 and is incremented with each tunnel that is created. The value of this object will wrap at 2,147,483,647.
jnxIpSecTunMonLocalGwAddrType	**jnxIpSecTunnelMonEntry 4**	IP address type of local gateway (endpoint) for the IPsec Phase 2 tunnel.
jnxIpSecTunMonLocalGwAddr	**jnxIpSecTunnelMonEntry 5**	IP address of local gateway (endpoint) for the IPsec Phase 2 tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 105: IPsec Phase 2 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxIpSecTunMonLocalProxyId	jnxIpSecTunnelMonEntry 6	Identifier for local end.
jnxIpSecTunMonRemoteProxyId	jnxIpSecTunnelMonEntry 7	Identifier for remote end.
jnxIpSecTunMonKeyType	jnxIpSecTunnelMonEntry 8	Type of key used by IPsec Phase 2 tunnel. It can be one of the following two types: • IKE-negotiated • Manually installed
jnxIpSecTunMonRemotePeerType	jnxIpSecTunnelMonEntry 9	Type of the remote peer gateway (endpoint). It can be one of the following two types: • Static (remote peer whose IP address is known beforehand) • Dynamic (remote peer whose IP address is not known beforehand)
jnxIpSecTunMonOutEncryptedBytes	jnxIpSecTunnelMonEntry 10	Number of bytes encrypted by this Phase 2 tunnel.
jnxIpSecTunMonOutEncryptedPkts	jnxIpSecTunnelMonEntry 11	Number of packets encrypted by this Phase 2 tunnel.
jnxIpSecTunMonInDecryptedBytes	jnxIpSecTunnelMonEntry 12	Number of bytes decrypted by this Phase 2 tunnel.
jnxIpSecTunMonInDecryptedPkts	jnxIpSecTunnelMonEntry 13	Number of packets decrypted by this Phase 2 tunnel.
jnxIpSecTunMonAHInBytes	jnxIpSecTunnelMonEntry 14	Number of incoming bytes authenticated using AH by this Phase 2 tunnel.
jnxIpSecTunMonAHInPkts	jnxIpSecTunnelMonEntry 15	Number of incoming packets authenticated using AH by this Phase 2 tunnel.
jnxIpSecTunMonAHOutBytes	jnxIpSecTunnelMonEntry 16	Number of outgoing bytes applied AH by this Phase 2 tunnel.
jnxIpSecTunMonAHOutPkts	jnxIpSecTunnelMonEntry 17	Number of outgoing packets applied AH by this Phase 2 tunnel.
jnxIpSecTunMonReplayDropPkts	jnxIpSecTunnelMonEntry 18	Number of packets dropped by this Phase 2 tunnel due to antireplay check failure.
jnxIpSecTunMonAhAuthFails	jnxIpSecTunnelMonEntry 19	Number of packets received by this Phase 2 tunnel that failed AH authentication.
jnxIpSecTunMonEspAuthFails	jnxIpSecTunnelMonEntry 20	Number of packets received by this Phase 2 tunnel that failed ESP authentication.
jnxIpSecTunMonDecryptFails	jnxIpSecTunnelMonEntry 21	Number of packets received by this Phase 2 tunnel that failed decryption.

Copyright © 2012, Juniper Networks, Inc.

Table 105: IPsec Phase 2 IKE Tunnel Table *(continued)*

Object	Object ID	Description
jnxIpSecTunMonBadHeaders	**jnxIpSecTunnelMonEntry 22**	Number of packets received by this Phase 2 tunnel that failed due to bad headers.
jnxIpSecTunMonBadTrailers	**jnxIpSecTunnelMonEntry 23**	Number of packets received by this Phase 2 tunnel that failed due to bad ESP trailers.
NOTE: The **jnxIpSecTunMonDroppedPkts** object is not supported in this release.		
jnxIpSecTunMonDroppedPkts	**jnxIpSecTunnelMonEntry 26**	Total number of dropped packets for this Phase 2 tunnel.

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 Security Association Table on page 411

IPsec Phase 2 Security Association Table

jnxIpSecSaMonTable, whose object ID is **{jnxIpSecFlowMonPhaseTwo 3}**, identifies the objects listed in Table 106 on page 412. The IPsec Phase 2 Security Association table identifies the structure (in terms of component SAs) of each active Phase 2 IPsec tunnel. This table contains an entry for each active and expiring SA and maps each entry in the active Phase 2 tunnel table (**ipSecTunTable**) into a number of entries in this table.

The SA contains the information negotiated by IKE. The SA is like a contract laying out the rules of the VPN connection for the duration of the SA. An SA is assigned a 32-bit number that, when used in conjunction with the destination IP address, uniquely identifies the SA. This number is called the Security Parameters Index (SPI).

IPsec SAs are unidirectional and are unique in each security protocol. A set of SAs is needed for a protected data pipe, one per direction per protocol.

Table 106: IPsec Phase 2 Security Association Table

Object	Object ID	Description
jnxIpSecSaMonEntry	**jnxIpSecSaMonTable 1**	Each entry contains the attributes associated with active and expiring IPsec Phase 2 SAs. Sequence of parameters: • **jnxIpSecSaMonIndex** • **jnxIpSecSaMonProtocol** • **jnxIpSecSaMonInSpi** • **jnxIpSecSaMonOutSpi** • **jnxIpSecSaMonType** • **jnxIpSecSaMonEncapMode** • **jnxIpSecSaMonLifeSize** • **jnxIpSecSaMonLifeTime** • **jnxIpSecSaMonActiveTime** • **jnxIpSecSaMonLifeSizeThreshold** (not supported in this release) • **jnxIpSecSaMonLifeTimeThreshold** • **jnxIpSecSaMonEncryptAlgo** • **jnxIpSecSaMonAuthAlgo** • **jnxIpSecSaMonState**
jnxIpSecSaMonIndex	**jnxIpSecSaMonEntry 1**	Index number, in the context of the IPsec tunnel **ipSecTunIndex**, of the SA represented by this table entry. The index number begins at 1 and is incremented with each SPI associated with an IPsec Phase 2 tunnel. The value of this object will wrap at 65535.
jnxIpSecSaMonProtocol	**jnxIpSecSaMonEntry 2**	Index number that represents the security protocol (AH, ESP or IPComp) for which this SA was set up.
jnxIpSecSaMonInSpi	**jnxIpSecSaMonEntry 3**	Value of the incoming SPI.
jnxIpSecSaMonOutSpi	**jnxIpSecSaMonEntry 4**	Value of the outgoing SPI.
jnxIpSecSaMonType	**jnxIpSecSaMonEntry 5**	Types of SAs that can be either manual or dynamic.
jnxIpSecSaMonEncapMode	**jnxIpSecSaMonEntry 6**	Encapsulation mode used by an IPsec Phase 2 tunnel.
jnxIpSecSaMonLifeSize	**jnxIpSecSaMonEntry 7**	Negotiated lifesize of the IPsec Phase 2 tunnel in kilobytes.
jnxIpSecSaMonLifeTime	**jnxIpSecSaMonEntry 8**	Negotiated lifetime of the IPsec Phase 2 tunnel in seconds.
jnxIpSecSaMonActiveTime	**jnxIpSecSaMonEntry 9**	Length of time the IPsec Phase 2 tunnel has been active in hundredths of seconds.
NOTE: The **jnxIpSecSaMonLifeSizeThreshold** object is not supported in this release.		
jnxIpSecSaMonLifeSizeThreshold	**jnxIpSecSaMonEntry 10**	SA lifesize refresh threshold in kilobytes.

Copyright © 2012, Juniper Networks, Inc.

Table 106: IPsec Phase 2 Security Association Table *(continued)*

Object	Object ID	Description
jnxIpSecSaMonLifeTimeThreshold	**jnxIpSecSaMonEntry 11**	SA lifetime refresh threshold in seconds.
jnxIpSecSaMonEncryptAlgo	**jnxIpSecSaMonEntry 12**	Encryption algorithm used to encrypt the packets that can be either **es-cbc** or **3des-cbc**.
jnxIpSecSaMonAuthAlgo	**jnxIpSecSaMonEntry 13**	Algorithm used for authentication of packets that can be **hmac-md5-96** or **hmac-sha1-96**.
jnxIpSecSaMonState	**jnxIpSecSaMonEntry 14**	This column represents the status of the SA represented by this table entry. If the status of the SA is **active**, the SA is ready for active use. The status **expiring** represents any of the various states that the SA transitions through before being purged.

Related Documentation

- IPsec Generic Flow Monitoring Object MIB on page 399
- Branch Tree Objects on page 400
- IPSec Generic Flow Monitoring Object MIB Textual Conventions on page 400
- Number of IKE Tunnels Currently Active on page 404
- IPsec Phase 1 IKE Tunnel Table on page 404
- IPsec Phase 2 IKE Tunnel Table on page 408

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 32

Interpreting the Enterprise-Specific IPsec Monitoring MIB

- IPSec Monitoring MIB on page 415
- jnxIkeTunnelTable on page 415
- jnxIPSecTunnelTable on page 418
- jnxIPSecSaTable on page 421

IPSec Monitoring MIB

The Juniper Networks enterprise-specific IPsec Monitoring MIB, whose object identifier is **{jnxMibs 22}**, provides operational and statistical information related to the IPsec and Internet Key Exchange (IKE) tunnels on Juniper Networks routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt .

For information about IPsec Monitoring MIB objects, see the following topics:

- jnxIkeTunnelTable on page 415
- jnxIPSecTunnelTable on page 418
- jnxIPSecSaTable on page 421

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxIkeTunnelTable

The IKE tunnel table (**jnxIkeTunnelTable**), whose object identifier is **{jnxIPSecPhaseOne 1}**, is used to monitor the IKE security associations established with the remote peers. The MIB variables in this table are used to display the IKE SA attributes and the SA statistics. There is one entry for each IKE SA present.

The key for this table is the combination of a service set name, remote gateway address, and the IKE tunnel index. The service set name is used from the **jnxSpSvcSetTable** which is implemented as part of the Services PIC MIB. The SNMP manager uses the

jnxSpSvcSetTable to get the service set name, and this information can then be used to query the **jnxIkeTunnelTable** for the given service set.

To get only IKE tunnels specific to a particular remote gateway in a service set, the SNMP manager can specify the corresponding service set name and the remote gateway address in the query.

The **jnxIkeTunnelEntry**, whose object identifier is **{jnxIkeTunnelTable 1}**, has 25 objects, which are listed in Table 107 on page 416. Each entry contains attributes associated with an active IPsec phase 1 IKE tunnel.

Table 107: jnxIkeTunnelTable

Object	Object Identifier	Description
jnxIkeTunIndex	**jnxIkeTunnelEntry 1**	Index for the table. The value of the index is a number that begins at 1 and is incremented with each tunnel that is created. When the index number reaches 2,147,483,647, the value wraps back to 1.
jnxIkeTunLocalRole	**jnxIkeTunnelEntry 2**	The role of the local peer identity. The role can be **initiator** or **responder**.
jnxIkeTunNegState	**jnxIkeTunnelEntry 3**	The state of the current negotiation. The state can be **matured** or **non matured**.
jnxIkeTunInitiatorCookie	**jnxIkeTunnelEntry 4**	Cookie generated by the peer that initiated the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunResponderCookie	**jnxIkeTunnelEntry 5**	Cookie generated by the peer responding to the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
jnxIkeTunLocalIdType	**jnxIkeTunnelEntry 6**	The type of local peer identity. A local peer can be identified by an IP address, a fully qualified domain name (FQDN), or a distinguished name.
jnxIkeTunLocalIdValue	**jnxIkeTunnelEntry 7**	The value of the local peer identity. • If the local peer type is an IP address, then this is the IP address used to identify the local peer. • If the local peer type is an FQDN(**if_fqdn**), then this is the FQDN of the remote peer. • If the local peer type is a distinguished name (**id_dn**), then this is the distinguished name of the local peer.
jnxIkeTunLocalGwAddrType	**jnxIkeTunnelEntry 8**	The IP address type of the local endpoint (gateway) for the IPsec phase 1 IKE tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 107: jnxIkeTunnelTable *(continued)*

Object	Object Identifier	Description
jnxIkeTunLocalGwAddr	jnxIkeTunnelEntry 9	The IP address of the local endpoint (gateway) for the IPsec phase 1 IKE tunnel.
jnxIkeTunLocalCertName	jnxIkeTunnelEntry 10	The name of the certificate used for authentication of the local tunnel endpoint. This object has a valid value only if the negotiated IKE authentication method is something other than a preshared key. If the IKE negotiation does not use certificates for authentication, the value is **NULL**.
jnxIkeTunRemoteIdType	jnxIkeTunnelEntry 11	The type of remote peer identity. A remote peer can be identified by an IP address, an FQDN, or a distinguished name.
jnxIkeTunRemoteIdValue	jnxIkeTunnelEntry 12	The value of the remote peer identity. • If the remote peer type is an IP address, then this is the IP address used to identify the remote peer. • If the remote peer type is an FQDN (**if_fqdn**), then this is the FQDN of the remote peer. • If the remote peer type is a distinguished name (**id_dn**), then this is the distinguished name of the remote peer.
jnxIkeTunRemoteGwAddrType	jnxIkeTunnelEntry 13	The IP address type of the remote gateway (endpoint) for the IPsec phase 1 IKE tunnel.
jnxIkeTunRemoteGwAddr	jnxIkeTunnelEntry 14	The IP address of the remote gateway (endpoint) for the IPsec phase 1 IKE tunnel.
jnxIkeTunNegoMode	jnxIkeTunnelEntry 15	The negotiation mode of the IPsec phase 1 IKE tunnel.
jnxIkeTunDiffHellmanGrp	jnxIkeTunnelEntry 16	The Diffie Hellman Group used in IPsec phase 1 IKE negotiations.
jnxIkeTunEncryptAlgo	jnxIkeTunnelEntry 17	The encryption algorithm used in IPsec phase 1 IKE negotiations.
jnxIkeTunHashAlgo	jnxIkeTunnelEntry 18	The hash algorithm used in IPsec phase 1 IKE negotiations.
jnxIkeTunAuthMethod	jnxIkeTunnelEntry 19	The authentication method used in IPsec phase 1 IKE negotiations.
jnxIkeTunLifeTime	jnxIkeTunnelEntry 20	The negotiated lifetime (in seconds) of the IPsec phase 1 IKE tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 107: jnxIkeTunnelTable *(continued)*

Object	Object Identifier	Description
jnxIkeTunActiveTime	**jnxIkeTunnelEntry 21**	The length of time (in hundredths of seconds) that the IPsec phase 1 IKE tunnel has been active.
jnxIkeTunInOctets	**jnxIkeTunnelEntry 22**	The total number of octets received by this IPsec phase 1 IKE security association.
jnxIkeTunInPkts	**jnxIkeTunnelEntry 23**	The total number of packets received by this IPsec phase 1 IKE security association.
jnxIkeTunOutOcets	**jnxIkeTunnelEntry 24**	The total number of octets sent by this IPsec phase 1 IKE security association.
jnxIkeTunOutPkts	**jnxIkeTunnelEntry 25**	The total number of octets sent by this IPsec phase 1 IKE security association.

Related Documentation

- IPSec Monitoring MIB on page 415
- jnxIPSecTunnelTable on page 418
- jnxIPSecSaTable on page 421

jnxIPSecTunnelTable

The IPsec phase 2 tunnel table **jnxIPSecTunnelTable)**, whose object identifier is **{jnxIPSecPhaseTwo 1}**, is used to monitor the IPsec phase 2 tunnel attributes along with the statistics for the tunnel. There is one entry for each tunnel to the peer security gateway. This table does not contain information about IPsec security associations (SAs) because multiple SAs can be present for each tunnel.

Similar to the IKE tunnel table (**jnxIkeTunnelTable**), the key of this table is a combination of the service set name, remote gateway address, and the IPsec tunnel index. This table can be queried just like the IKE tunnel table.

To get only IPsec tunnels specific to a particular remote gateway in a service set, the SNMP manager can specify the corresponding service set name and the remote gateway address in the query.

The **jnxIPSecTunnelEntry**, whose object identifier is **{jnxIPSecTunnelTable 1}**, has 27 objects, which are listed in Table 108 on page 419. Each entry contains attributes associated with an active IPsec phase 2 tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 108: jnxIPSecTunnelTable

Object	Object Identifier	Description
jnxIPSecTunIndex	jnxIPSecTunnelEntry 1	Index for the table. The value of the index is a number that begins at 1 and is incremented with each tunnel that is created. When the index number reaches 2,147,483,647, the value wraps back to 1.
jnxIPSecRuleName	jnxIPSecTunnelEntry 2	The name of the rule defined in the IPsec configuration.
jnxIPSecTermName	jnxIPSecTunnelEntry 3	The name of the term configured under the IPsec rule.
jnxIPSecTunLocalGwAddrType	jnxIPSecTunnelEntry 4	The IP address type of the local gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunLocalGwAddr	jnxIPSecTunnelEntry 5	The IP address of the local gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunRemoteGwAddrType	jnxIPSecTunnelEntry 6	The IP address type of the remote gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunRemoteGwAddr	jnxIPSecTunnelEntry 7	The IP address of the remote gateway (endpoint) for the IPsec phase 2 tunnel.
jnxIPSecTunLocalProxyId	jnxIPSecTunnelEntry 8	The identifier for the local endpoint.
jnxIPSecTunRemoteProxyId	jnxIPSecTunnelEntry 9	The identifier for the remote endpoint.
jnxIPSecTunKeyType	jnxIPSecTunnelEntry 10	The type of key used by the IPsec phase 2 tunnel. The key type can be **IKE negotiated** or **Manually installed**.
jnxIPSecRemotePeerType	jnxIPSecTunnelEntry 11	The type of the remote peer gateway (endpoint). If the remote peer's IP address is known beforehand, the type is **static**. If the IP address is not known beforehand, the type is **dynamic**.
jnxIPSecTunMtu	jnxIPSecTunnelEntry 12	The maximum transmission unit (MTU) value of the IPsec phase 2 tunnel.
jnxIPSecTunOutEncryptedBytes	jnxIPSecTunnelEntry 13	The number of bytes encrypted by the IPsec phase 2 tunnel.
jnxIPSecTunOutEncryptedPkts	jnxIPSecTunnelEntry 14	The number of packets encrypted by the IPsec phase 2 tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 108: jnxIPSecTunnelTable *(continued)*

Object	Object Identifier	Description
jnxIPSecTunInDecryptedBytes	jnxIPSecTunnelEntry 15	The number of bytes decrypted by the IPsec phase 2 tunnel.
jnxIPSecTunInDecryptedPkts	jnxIPSecTunnelEntry 16	The number of packets decrypted by the IPsec phase 2 tunnel.
jnxIPSecTunAHInBytes	jnxIPSecTunnelEntry 17	The number of incoming bytes authenticated using the authentication header (AH) by the IPsec phase 2 tunnel.
jnxIPSecTunAHInPkts	jnxIPSecTunnelEntry 18	The number of incoming packets authenticated using the authentication header (AH) by the IPsec phase 2 tunnel.
jnxIPSecTunAHOutBytes	jnxIPSecTunnelEntry 19	The number of outgoing bytes on the IPsec phase 2 tunnel where the AH is applied.
jnxIPSecTunHAOutPkts	jnxIPSecTunnelEntry 20	The number of outgoing packets on the IPsec phase 2 tunnel where the AH is applied.
jnxIPSecTunReplayDropPkts	jnxIPSecTunnelEntry 21	The number of packets dropped by the IPsec phase 2 tunnel because of an anti-replay check failure.
jnxIPSecTunAhAuthFails	jnxIPSecTunnelEntry 22	The number of packets received by the IPsec phase 2 tunnel that failed AH authentication.
jnxIPSecTunEspAuthFails	jnxIPSecTunnelEntry 23	The number of packets received by this IPsec phase 2 tunnel that failed ESP authentication.
jnxIPSecTunDecryptFails	jnxIPSecTunnelEntry 24	The number of packets received by this IPsec phase 2 tunnel that failed decryption.
jnxIPSecTunBadHeaders	jnxIPSecTunnelEntry 25	The number of packets received by this IPsec phase 2 tunnel that failed because of bad headers.
jnxIPSecTunBadTrailers	jnxIPSecTunnelEntry 26	The number of packets received by this IPsec phase 2 tunnel that failed because of bad ESP trailers.
jnxIPSecTunDroppedPkts	jnxIPSecTunnelEntry 27	The total number of packets dropped from this IPsec phase 2 tunnel.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- IPSec Monitoring MIB on page 415
- jnxIkeTunnelTable on page 415
- jnxIPSecSaTable on page 421

jnxIPSecSaTable

The IPsec phase 2 security association table (**jnxIPSecSaTable**), whose object identifier is **{jnxIPSecPhaseTwo 2}**, is used to monitor the IPsec SAs present for each tunnel in the IPsec tunnel table (**jnxIPSecTunnelTable**). More than one pair of SAs can be present for each of the IPsec tunnels.

The key for this table is a combination of a service set name, remote gateway address, IPsec tunnel index, and the SA index. While the IPsec tunnel table is queried using the service set name, the SA table can be queried for the IPsec tunnel using the service set name, remote gateway address, and the IPsec tunnel index.

The **jnxIPSecSaEntry**, whose object identifier is **{jnxIPSecSaTable 1}**, has 16 objects, which are listed in Table 109 on page 421. Each entry contains SA components for an active IPsec phase 2 tunnel.

Table 109: jnxIPSecSaTable

Object	Object Identifier	Description
jnxIpSecSaProtocol	**jnxIpSecSaEntry** 1	The index represents the security protocol (AH, ESP, or IPComp) for which the SA was created.
jnxIpSecSaIndex	**jnxIpSecSaEntry 2**	The index (in the context of the IPsec tunnel) for the SA. The value of the index is a number that begins at 1 and is incremented with each security parameter index (SPI) associated with an IPsec phase 2 tunnel. When the index number reaches 2,147,483,647, the value wraps back to 1.
jnxIpSecSaInSpi	**jnxIpSecSaEntry 3**	The value of the incoming SPI.
jnxIpSecSaOutSpi	**jnxIpSecSaEntry 4**	The value of the outgoing SPI.
jnxIpSecSaInAuxSpi	**jnxIpSecSaEntry 5**	The value of the incoming auxiliary SPI. This object is valid for AH and ESP bundles.
jnxIpSecSaOutAuxSpi	**jnxIpSecSaEntry 6**	The value of the outgoing auxiliary SPI. This object is valid for AH and ESP bundles.
jnxIpSecSaType	**jnxIpSecSaEntry 7**	The type of SA (**manual** or **dynamic**).
jnxIpSecSaEncapMode	**jnxIpSecSaEntry 8**	The encapsulation mode used by the IPsec phase 2 tunnel.

Copyright © 2012, Juniper Networks, Inc.

Table 109: jnxIPSecSaTable *(continued)*

Object	Object Identifier	Description
jnxIpSecSaLifeSize	**jnxIpSecSaEntry 9**	The negotiated size (in kilobytes) of the IPsec phase 2 tunnel.
jnxIpSecSaLifeTime	**jnxIpSecSaEntry 10**	The negotiated **lifetime** (in seconds) of the IPsec phase 2 tunnel.
jnxIpSecSaActiveTime	**jnxIpSecSaEntry 11**	The number of seconds the IPsec phase 2 tunnel has been active.
jnxIpSecSaLifeSizeThreshold	**jnxIpSecSaEntry 12**	The refresh threshold (in kilobytes) of the SA size.
jnxIpSecSaLifeTimeThreshold	**jnxIpSecSaEntry 13**	The refresh threshold (in seconds) of the SA lifetime.
jnxIpSecSaEncryptAlgo	**jnxIpSecSaEntry 14**	The algorithm used to encrypt the packets (**es-cbc** or **3des-cbc**).
jnxIpSecSaAuthAlgo	**jnxIpSecSaEntry 15**	The algorithm used to authenticate the packets (**hmac-md5-96** or **hmac-sha1-96**).
jnxIpSecSaState	**jnxIpSecSaEntry 16**	The status of the SA. Status can be **active** (ready for active use) or **expiring** (any state an SA goes through before being purged).

Related Documentation

- IPSec Monitoring MIB on page 415
- jnxIkeTunnelTable on page 415
- jnxIPSecTunnelTable on page 418

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 33

Interpreting the Enterprise-Specific IPv4 MIB

- IPv4 MIB on page 423
- jnxIpv4AddrTable on page 423

IPv4 MIB

The Juniper Networks enterprise-specific IPv4 MIB, whose object identifier is **{jnxMibs12}**, functions as an extension of the **ifTable** defined in RFC 1573, *IF MIB*, and defines the branches for IPV4 configuration.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-ipv4.txt .

For information about the enterprise-specific IPv4 MIB objects, see the following topic:

- jnxIpv4AddrTable on page 423

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxIpv4AddrTable

The **jnxIpv4AddrTable** defines the **jnxIpv4AddrEntry** and its attributes. Each **jnxIpv4AddrEntry** contains the objects listed in Table 110 on page 423.

Table 110: jnxIpv4AddrTable

Object	Object ID	Description
jnxIpv4AdEntIfIndex	**jnxIpv4AddrEntry 1**	A unique index value that identifies the interface with which a particular entry is associated. An interface identified by a particular value of **jnxIpv4AdEntIfIndex** is the same as the interface that is identified by the same value of **ifIndex** as defined in RFC 1573.
jnxIpv4AdEntAddr	**jnxIpv4AddrEntry 2**	The IP address of the interface with which the address information stored in this entry is associated.

Table 110: jnxIpv4AddrTable *(continued)*

Object	Object ID	Description
jnxIpv4AdEntNetMask	**jnxIpv4AddrEntry 3**	The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0.
jnxIpv4AdEntBcastAddr	**jnxIpv4AddrEntry 4**	The least significant bit in the IP broadcast address used for sending datagrams on the logical interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value is 1. This value applies to both the subnet and network broadcast addresses used by the entry on the logical interface.
jnxIpv4AdEntReasmMaxSize	**jnxIpv4AddrEntry 5**	The size of the largest IP datagram that this entry can reassemble from the incoming fragmented IP datagrams received on the interface.

Related Documentation

- IPv4 MIB on page 423

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 34

Interpreting the Enterprise-Specific IPv6 MIB

- IPv6 MIB on page 425
- jnxIpv6GlobalStats on page 425
- jnxIcmpv6GlobalStats on page 428

IPv6 MIB

The Juniper Networks enterprise-specific IPv6 MIB, whose object identifier is **{jnxMibs 11}**, functions as an extension of the **ifTable** defined in RFC 2465, *IF MIB*, and defines the branches of IPv6 configuration.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-ipv6.txt .

For more information about the enterprise-specific IPv6 MIB objects, see the following topics:

- jnxIpv6GlobalStats on page 425
- jnxIcmpv6GlobalStats on page 428

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxIpv6GlobalStats

The **jnxIpv6GlobalStats**, whose object ID is **{jnxIpv6Stats 1}**, represents the **jnxIpv6GlobalStats** branch of IPv6 stats. Table 111 on page 425 identifies objects in the **jnxIpv6GlobalStats** branch.

Table 111: jnxIpv6GlobalStats

Object	Object ID	Description
jnxIpv6StatsReceives	**jnxIpv6GlobalStats 1**	The total number of IPv6 packets received.

Table 111: jnxIpv6GlobalStats *(continued)*

Object	Object ID	Description
jnxIpv6StatsTooShorts	jnxIpv6GlobalStats 2	The total number of IPv6 packets received with size smaller than minimum.
jnxIpv6StatsTooSmalls	jnxIpv6GlobalStats 3	The total number of IPv6 packets with data size less than data length.
jnxIpv6StatsBadOptions	jnxIpv6GlobalStats 4	The total number of IPv6 packets encountering errors in option processing.
jnxIpv6StatsBadVersions	jnxIpv6GlobalStats 5	The total number of IPv6 packets with incorrect version number.
jnxIpv6StatsFragments	jnxIpv6GlobalStats 6	The total number of packet fragments received.
jnxIpv6StatsFragDrops	jnxIpv6GlobalStats 7	The total number of fragments dropped (duplicates or out of place).
jnxIpv6StatsFragTimeOuts	jnxIpv6GlobalStats 8	The total number of fragments dropped due to timeout.
jnxIpv6StatsFragOverFlows	jnxIpv6GlobalStats 9	The total number of fragments that exceeded limit.
jnxIpv6StatsReasmOKs	jnxIpv6GlobalStats 10	The total number of IPv6 packets reassembled successfully.
jnxIpv6StatsDelivers	jnxIpv6GlobalStats 11	The total number of IPv6 packets delivered to the upper layers.
jnxIpv6StatsForwards	jnxIpv6GlobalStats 12	The total number of IPv6 packets forwarded.
jnxIpv6StatsUnreachables	jnxIpv6GlobalStats 13	The total number of IPv6 packets received for unreachable destinations.
jnxIpv6StatsRedirects	jnxIpv6GlobalStats 14	The total number of IPv6 packets forwarded on the same net as received.
jnxIpv6StatsOutRequests	jnxIpv6GlobalStats 15	The total number of IPv6 packets generated by the current host.
jnxIpv6StatsRawOuts	jnxIpv6GlobalStats 16	The total number of IPv6 packets sent with fabricated IPv6 headers.
jnxIpv6StatsOutDiscards	jnxIpv6GlobalStats 17	The total number of output IPv6 packets dropped due to no buffers and so on.
jnxIpv6StatsOutNoRoutes	jnxIpv6GlobalStats 18	The total number of output IPv6 packets discarded because no route was found.
jnxIpv6StatsOutFragOKs	jnxIpv6GlobalStats 19	The total number of output IPv6 packets fragmented.
jnxIpv6StatsOutFragCreates	jnxIpv6GlobalStats 20	The total number of output IPv6 packets that have been generated as a result of fragmentation at the output interface.

Copyright © 2012, Juniper Networks, Inc.

Table 111: jnxIpv6GlobalStats *(continued)*

Object	Object ID	Description
jnxIpv6StatsOutFragFails	jnxIpv6GlobalStats 21	The total number of output IPv6 packets that cannot be fragmented.
jnxIpv6StatsBadScopes	jnxIpv6GlobalStats 22	The total number of IPv6 packets that violate scope rules.
jnxIpv6StatsNotMcastMembers	jnxIpv6GlobalStats 23	The total number of IPv6 multicast packets that the users do not join.
jnxIpv6StatsHdrNotContinuous	jnxIpv6GlobalStats 24	The total number of IPv6 packets whose headers are not continuous.
jnxIpv6StatsNoGifs	jnxIpv6GlobalStats 25	The total number of tunneling IPv6 packets that canot find graphic interchange format (GIF).
jnxIpv6StatsTooManyHdrs	jnxIpv6GlobalStats 26	The total number of IPv6 packets discarded due to numerous headers.
jnxIpv6StatsForwCacheHits	jnxIpv6GlobalStats 27	The total number of forward cache hits.
jnxIpv6StatsForwCacheMisses	jnxIpv6GlobalStats 28	The total number of forward cache misses.
jnxIpv6StatsOutDeadNextHops	jnxIpv6GlobalStats 29	The total number of output IPv6 packets destined to dead next hops.
jnxIpv6StatsOptRateDrops	jnxIpv6GlobalStats 30	The total number of IPv6 option packets dropped due to rate limits.
jnxIpv6StatsMCNoDests	jnxIpv6GlobalStats 31	The total number of IPv6 Multicast packets dropped due to no destination.

Table 112 on page 427 identifies the count of all the other IPv6 next header types received.

Table 112: Count of All Other IPv6 Next Header Types Received

Object	Object ID	Description
jnxIpv6StatsInHopByHops	jnxIpv6GlobalStats 32	The total number of IPv6 packets received with a Hop by Hop next header.
jnxIpv6StatsInIcmps	jnxIpv6GlobalStats 33	The total number of IPv6 packets received with an ICMP(v4) next header.
jnxIpv6StatsInIgmps	jnxIpv6GlobalStats 34	The total number of IPv6 packets received with an IGMP next header.
jnxIpv6StatsInIps	jnxIpv6GlobalStats 35	The total number of IPv6 packets received with an IPv4 next header.
jnxIpv6StatsInTcps	jnxIpv6GlobalStats 36	The total number of IPv6 packets received with a TCP next header.

Copyright © 2012, Juniper Networks, Inc.

Table 112: Count of All Other IPv6 Next Header Types Received *(continued)*

Object	Object ID	Description
jnxIpv6StatsInUdps	jnxIpv6GlobalStats 37	The total number of IPv6 packets received with a UDP next header.
jnxIpv6StatsInIdps	jnxIpv6GlobalStats 38	The total number of IPv6 packets received with an xns IDP next header.
jnxIpv6StatsInTps	jnxIpv6GlobalStats 39	The total number of IPv6 packets received with a TP4 next header.
jnxIpv6StatsInIpv6s	jnxIpv6GlobalStats 40	The total number of IPv6 packets with an IPv6 next header.
jnxIpv6StatsInRoutings	jnxIpv6GlobalStats 41	The total number of IPv6 packets with an IPv6 Routing next header.
jnxIpv6StatsInFrags	jnxIpv6GlobalStats 42	The total number of IPv6 packets with an IPv6 Fragment next header.
jnxIpv6StatsInEsps	jnxIpv6GlobalStats 43	The total number of IPv6 packets with an IPv6 ESP next header.
jnxIpv6StatsInAhs	jnxIpv6GlobalStats 44	The total number of IPv6 packets with an IPv6 AH next header.
jnxIpv6StatsInIcmpv6s	jnxIpv6GlobalStats 45	The total number of IPv6 packets with an ICMPv6 next header.
jnxIpv6StatsInNoNhs	jnxIpv6GlobalStats 46	The total number of IPv6 packets with no next header.
jnxIpv6StatsInDestOpts	jnxIpv6GlobalStats 47	The total number of IPv6 packets with an IPv6 Destination Option next header.
jnxIpv6StatsInIsoIps	jnxIpv6GlobalStats 48	The total number of IPv6 packets with an ISO CLNP next header.
jnxIpv6StatsInOspfs	jnxIpv6GlobalStats 49	The total number of IPv6 packets with an OSPF next header.
jnxIpv6StatsInEths	jnxIpv6GlobalStats 50	The total number of IPv6 packets with an Ethernet next header.
jnxIpv6StatsInPims	jnxIpv6GlobalStats 51	The total number of IPv6 packets with a PIM next header.

Related Documentation

- IPv6 MIB on page 425
- jnxIcmpv6GlobalStats on page 428

jnxIcmpv6GlobalStats

The **jnxIcmpv6GlobalStats**, whose object ID is **{jnxIpv6Stats 2}**, represents the jnxIcmpv6GlobalStats branch of IPv6 stats. Table 113 on page 429 identifies objects in the jnxIcmpv6GlobalStats branch.

Copyright © 2012, Juniper Networks, Inc.

Table 113: jnxIcmpv6GlobalStats

Object	Object ID	Description
jnxIcmpv6StatsErrors	jnxIcmpv6GlobalStats 1	The total number of calls to **icmp_error**.
jnxIcmpv6StatsCantErrors	jnxIcmpv6GlobalStats 2	The total number of errors that are not generated if the old message was an ICMP error.
jnxIcmpv6StatsTooFreqs	jnxIcmpv6GlobalStats 3	The total number of errors that are not generated due to rate limitations.
jnxIcmpv6StatsBadCodes	jnxIcmpv6GlobalStats 4	The total number of messages with bad code fields.
jnxIcmpv6StatsTooShorts	jnxIcmpv6GlobalStats 5	The total number of messages less than the minimum length.
jnxIcmpv6StatsBadChecksums	jnxIcmpv6GlobalStats 6	The total number of messages with bad checksums.
jnxIcmpv6StatsBadLenths	jnxIcmpv6GlobalStats 7	The total number of messages with bad length.
jnxIcmpv6StatsNoRoutes	jnxIcmpv6GlobalStats 8	The total number of “no route” messages generated.
jnxIcmpv6StatsAdminProhibits	jnxIcmpv6GlobalStats 9	The total number of “administratively prohibited” messages generated.
jnxIcmpv6StatsBeyondScopes	jnxIcmpv6GlobalStats 10	The total number of “beyond scope” messages generated.
jnxIcmpv6StatsAddrUnreachs	jnxIcmpv6GlobalStats 11	The total number of “address unreachable” messages generated.
jnxIcmpv6StatsPortUnreachs	jnxIcmpv6GlobalStats 12	The total number of “port unreachable” messages generated.
jnxIcmpv6StatsTooBigs	jnxIcmpv6GlobalStats 13	The total number of “packet too big” messages generated.
jnxIcmpv6StatsExceedTrans	jnxIcmpv6GlobalStats 14	The total number of “time exceed transit” messages generated.
jnxIcmpv6StatsExceedReasms	jnxIcmpv6GlobalStats 15	The total number of “time exceed reassembly” messages generated.
jnxIcmpv6StatsBadHdrFields	jnxIcmpv6GlobalStats 16	The total number of “erroneous header field” messages generated.
jnxIcmpv6StatsBadNextHdrs	jnxIcmpv6GlobalStats 17	The total number of “unrecognized next header” messages generated.
jnxIcmpv6StatsBadOptions	jnxIcmpv6GlobalStats 18	The total number of “unrecognized option” messages generated.
jnxIcmpv6StatsRedirects	jnxIcmpv6GlobalStats 19	The total number of “redirect” messages generated.

Copyright © 2012, Juniper Networks, Inc.

Table 113: jnxIcmpv6GlobalStats *(continued)*

Object	Object ID	Description
jnxIcmpv6StatsOthers	jnxIcmpv6GlobalStats 20	The total number of generated message types other than the ones previously listed.
jnxIcmpv6StatsResponses	jnxIcmpv6GlobalStats 21	The total number of message responses generated.
jnxIcmpv6StatsExcessNDOptions	jnxIcmpv6GlobalStats 22	The total number of messages processed with too many ND options.

Table 114 on page 430 identifies the count of all the other ICMPv6 message types received and transmitted.

Table 114: Count of All Other ICMPv6 Message Types Received and Transmitted

Object	Object ID	Description
jnxIcmpv6StatsInUnreachables	jnxIcmpv6GlobalStats 23	The total number of unreachable messages received.
jnxIcmpv6StatsInPktTooBigs	jnxIcmpv6GlobalStats 24	The total number of “packet too big” messages received.
jnxIcmpv6StatsInTimeExceeds	jnxIcmpv6GlobalStats 25	The total number of “time exceeded” messages received.
jnxIcmpv6StatsInParamProbs	jnxIcmpv6GlobalStats 26	The total number of “parameter problem” messages received.
jnxIcmpv6StatsInEchoReqs	jnxIcmpv6GlobalStats 27	The total number of “echo request” messages received.
jnxIcmpv6StatsInEchoReplies	jnxIcmpv6GlobalStats 28	The total number of “echo reply” messages received.
jnxIcmpv6StatsInMLQueries	jnxIcmpv6GlobalStats 29	The total number of “multicast listener query” messages received.
jnxIcmpv6StatsInMLReports	jnxIcmpv6GlobalStats 30	The total number of “multicast listener report” messages received.
jnxIcmpv6StatsInMLDones	jnxIcmpv6GlobalStats 31	The total number of “multicast listener done” messages received.
jnxIcmpv6StatsInRtrSolicits	jnxIcmpv6GlobalStats 32	The total number of “router solicitation” messages received.
jnxIcmpv6StatsInRtrAdvs	jnxIcmpv6GlobalStats 33	The total number of “router advertisement” messages received.
jnxIcmpv6StatsInNbrSolicits	jnxIcmpv6GlobalStats 34	The total number of “neighbor solicitation” messages received.
jnxIcmpv6StatsInNbrAdvs	jnxIcmpv6GlobalStats 35	The total number of “neighbor advertisement” messages received.
jnxIcmpv6StatsInRedirects	jnxIcmpv6GlobalStats 36	The total number of “redirect” messages received.

Copyright © 2012, Juniper Networks, Inc.

Table 114: Count of All Other ICMPv6 Message Types Received and Transmitted *(continued)*

Object	Object ID	Description
jnxIcmpv6StatsInRtrRenumbers	jnxIcmpv6GlobalStats 37	The total number of “router renumber” messages received.
jnxIcmpv6StatsInNIReqs	jnxIcmpv6GlobalStats 38	The total number of “node information request” messages received.
jnxIcmpv6StatsInNIReplies	jnxIcmpv6GlobalStats 39	The total number of “node information report” messages received.
jnxIcmpv6StatsOutUnreachables	jnxIcmpv6GlobalStats 40	The total number of “unreachable” messages transmitted.
jnxIcmpv6StatsOutPktTooBigs	jnxIcmpv6GlobalStats 41	The total number of “packet too big” messages to be transmitted. Some of the icmp messages are not sent out (problems with rate limits, source addresses, and so on). **jnxIcmpv6StatsTooBigs** represents the number of 'too big packet' messages that actually goes out.
jnxIcmpv6StatsOutTimeExceeds	jnxIcmpv6GlobalStats 42	The total number of “time exceeded” messages transmitted.
jnxIcmpv6StatsOutParamProbs	jnxIcmpv6GlobalStats 43	The total number of “parameter problem” messages transmitted.
jnxIcmpv6StatsOutEchoReqs	jnxIcmpv6GlobalStats 44	The total number of “echo request” messages transmitted.
jnxIcmpv6StatsOutEchoReplies	jnxIcmpv6GlobalStats 45	The total number of “echo reply” messages transmitted.
jnxIcmpv6StatsOutMLQueries	jnxIcmpv6GlobalStats 46	The total number of “multicast listener query” messages transmitted.
jnxIcmpv6StatsOutMLReports	jnxIcmpv6GlobalStats 47	The total number of “multicast listener report” messages transmitted.
jnxIcmpv6StatsOutMLDones	jnxIcmpv6GlobalStats 48	The total number of “multicast listener done” messages transmitted.
jnxIcmpv6StatsOutRtrSolicits	jnxIcmpv6GlobalStats 49	The total number of “router solicitation” messages transmitted.
jnxIcmpv6StatsOutRtrAdvs	jnxIcmpv6GlobalStats 50	The total number of “router advertisement” messages transmitted.
jnxIcmpv6StatsOutNbrSolicits	jnxIcmpv6GlobalStats 51	The total number of “neighbor solicitation” messages transmitted.
jnxIcmpv6StatsOutNbrAdvs	jnxIcmpv6GlobalStats 52	The total number of “neighbor advertisement” messages transmitted.
jnxIcmpv6StatsOutRedirects	jnxIcmpv6GlobalStats 53	The total number of “redirect” messages transmitted.

Copyright © 2012, Juniper Networks, Inc.

Table 114: Count of All Other ICMPv6 Message Types Received and Transmitted *(continued)*

Object	Object ID	Description
jnxIcmpv6StatsOutRtrRenumbers	jnxIcmpv6GlobalStats 54	The total number of “router renumber” messages transmitted.
jnxIcmpv6StatsOutNIReqs	jnxIcmpv6GlobalStats 55	The total number of “node information request” messages transmitted.
jnxIcmpv6StatsOutNIReplies	jnxIcmpv6GlobalStats 56	The total number of “node information report” messages transmitted.

Related Documentation

- IPv6 MIB on page 425
- jnxIpv6GlobalStats on page 425

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 35

Interpreting the Enterprise-Specific License MIB

- License MIB on page 433
- jnxLicenseInstallTable on page 434
- Supported Features and Configuration Parameters for Licensing on page 434
- Juniper Networks Enterprise-Specific License MIB Notifications on page 435

License MIB

The Juniper Networks enterprise-specific License MIB, whose object identifier is **{jnxMibs 63 }**, extends SNMP support to licensing information in the Junos OS.

The License MIB stores licensing-related information, such as licensed features installed on the device, licenses that are about to expire or have expired, licenses that are in use, the number of users accessing a licensed feature, and so on. It enables users to access licensing-related information using SNMP **get** and **get-next** commands. The License MIB also supports SNMP traps that alert users when a license is about to expire or has expired, when the number of users exceeds the maximum number specified in the license, and so on.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-license.txt .

For information about the enterprise-specific License MIB objects and traps, see the following topics:

- jnxLicenseInstallTable on page 434
- Supported Features and Configuration Parameters for Licensing on page 434
- Juniper Networks Enterprise-Specific License MIB Notifications on page 435

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLicenseInstallTable

The **jnxLicenseInstallTable**, whose object identifier is **{jnxLicenseInstallObjects 1}**, contains license-related information for the features installed on the device. The license-related information for installed features is stored in **jnxLicenseInstallEntry**.

Each **jnxLicenseInstallEntry**, whose object identifier is **{jnxLicenseInstallTable 1}**, contains the objects listed in Table 115 on page 434.

Table 115: jnxLicenseInstallEntry

Object	Object Identifier	Description
jnxLicenseId	**jnxLicenseInstallEntry 1**	Contains the license ID for the installed feature. It is also the index for **jnxLicenseInstallEntry**.
jnxLicenseVersion	**jnxLicenseInstallEntry 2**	Contains information about the license version.
jnxLicenseDeviceId	**jnxLicenseInstallEntry 3**	Contains the device ID for the license.
jnxLicenseType	**jnxLicenseInstallEntry 4**	Indicates the license type. Possible values are permanent, time-based, and invalid.
jnxLicenseKeys	**jnxLicenseInstallEntry 5**	Contains the license key.
jnxLicenseUsed	**jnxLicenseInstallEntry 6**	Indicates whether the license is in use (1) or not (0).
jnxLicenseInstalled	**jnxLicenseInstallEntry 7**	Contains details of the installed license.
jnxLicenseNeeded	**jnxLicenseInstallEntry 8**	Indicates whether the license needs to be installed.

Related Documentation

- License MIB on page 433
- Supported Features and Configuration Parameters for Licensing on page 434
- Juniper Networks Enterprise-Specific License MIB Notifications on page 435

Supported Features and Configuration Parameters for Licensing

This topic contains the following sections:

- jnxLicenseFeatureListTable on page 434
- License Configuration Parameters on page 435

jnxLicenseFeatureListTable

The **jnxLicenseFeatureListTable**, whose object identifier is **{jnxLicenseInstallObjects 2}**, contains a list of features that support licensing. Each **jnxLicenseFeatureListEntry**, whose object identifier is **{jnxLicenseFeatureListTable 1}**, contains the objects listed in Table 116 on page 435.

Copyright © 2012, Juniper Networks, Inc.

Table 116: jnxLicenseFeatureListEntry

Object	Object Identifier	Description
jnxLicenseFeatureId	**jnxLicenseFeatureListEntry 1**	Contains the feature ID that is the index of the **jnxLicenseFeatureListEntry**.
jnxLicenseFeatureName	**jnxLicenseFeatureListEntry 2**	Contains the name of the feature.
jnxLicenseFeatureDescr	**jnxLicenseFeatureListEntry 3**	Contains a description of the feature.
jnxLicenseFeatureLicenseId	**jnxLicenseFeatureListEntry 4**	Contains the feature license ID (from the **jnxLicenseInstallTable**) that maps the feature to the license information.

License Configuration Parameters

The license MIB also contains the following objects that provide information about license configuration:

- **jnxLicenseRenewBeforExpiration**, whose object identifier is **{jnxLicenseSettings 1}**, indicates the number of days left before the license expires.
- **jnxLicenseRenewInterval**, whose object identifier is **{jnxLicenseSettings 2}**, indicates the license checking interval in hours.
- **jnxLicenseiAutoUpdate**, whose object identifier is **{jnxLicenseSettings 3}**, contains the auto-update URL of a license server.
- **jnxLicenseTraceOptions**, whose object identifier is **{jnxLicenseSettings 4}**, indicates the traceoption settings for the license. Possible values are all (1), events (2), and configuration (3).

Related Documentation

- License MIB on page 433
- jnxLicenseInstallTable on page 434
- Juniper Networks Enterprise-Specific License MIB Notifications on page 435

Juniper Networks Enterprise-Specific License MIB Notifications

Table 117 on page 435 lists the license MIB notifications.

Table 117: License MIB Notifications

Notification	Object Identifier	Description
jnxLicenseGraceExpired	**jnxLicenseNotifications 1**	Generated when the grace period for renewing the license for the feature (identified by **jnxLicenseFeatureName**) has expired.

Table 117: License MIB Notifications *(continued)*

Notification	Object Identifier	Description
jnxLicenseGraceAboutToExpire	**jnxLicenseNotifications 2**	Generated when the grace period for renewing the license for the feature (identified by **jnxLicenseFeatureName**) is about to expire.
jnxLicenseAboutToExpire	**jnxLicenseNotifications 3**	Generated when the license for the feature (identified by **jnxLicenseFeatureName**) is about to expire.
jnxLicenseInfringeCumulative	**jnxLicenseNotifications 4**	Generated when the use of the feature (identified by **jnxLicenseFeatureName**) exceeds the maximum limit specified in the license.
jnxLicenseInfringeSingle	**jnxLicenseNotifications 5**	Generated when the license for the feature (identified by **jnxLicenseFeatureName**) is invalid; that is, either expired or not available.

Related Documentation

- License MIB on page 433
- jnxLicenseInstallTable on page 434
- Supported Features and Configuration Parameters for Licensing on page 434
- Standard SNMP Traps Supported on Devices Running Junos OS on page 113
- Juniper Networks Enterprise-Specific SNMP Traps on page 125

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 36

Interpreting the Enterprise-Specific Logical Systems (LSYS) MIBs

- LSYS Security Profile on page 437
- LSYS Zone MIB on page 438
- LSYS Scheduler MIB on page 440
- LSYS Policy MIB on page 443
- LSYS Policy with Count MIB on page 445
- LSYS Flow Gate MIB on page 448
- LSYS Flow Session MIB on page 450
- LSYS AUTH Entry MIB on page 452
- LSYS NAT Source Pool MIB on page 454
- LSYS NAT Destination Pool MIB on page 456
- LSYS NAT Source PAT Address MIB on page 459
- LSYS NAT Source No PAT Address MIB on page 461
- LSYS NAT Source Rule MIB on page 463
- LSYS NAT Destination Rule MIB on page 466
- LSYS NAT Static Rule MIB on page 468
- LSYS NAT Cone Bind MIB on page 471
- LSYS NAT Port Overloading IP Number MIB on page 473
- LSYS CPU MIB on page 475

LSYS Security Profile

- jnxLsysSecurityProfile on page 437

jnxLsysSecurityProfile

The Juniper Networks enterprise-specific LSYS Security Profile MIB defines a list of MIBs for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt.

For information about the logical system security profile, see the following topics:

- jnxLsysSpZoneMIB on page 438
- jnxLsysSpSchedulerMIB on page 441
- jnxLsysSpPolicyMIB on page 443
- jnxLsysSpPolicywcntSummary on page 447
- jnxLsysSpFlowgateMIB on page 448
- jnxLsysSpFlowsessMIB on page 450
- jnxLsysSpAuthentryMIB on page 452
- jnxLsysSpNATsrcpoolMIB on page 454
- jnxLsysSpNATdstpoolMIB on page 457
- jnxLsysSpNATsrcpatadMIB on page 459
- jnxLsysSpNATsrcnopatadMIB on page 461
- jnxLsysSpNATsrcruleMIB on page 463
- jnxLsysSpNATdstruleMIB on page 466
- jnxLsysSpNATstaticruleMIB on page 468
- jnxLsysSpNATconebindMIB on page 471
- jnxLsysSpNATpoipnumMIB on page 473
- jnxLsysSpCPUMIB on page 475

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

LSYS Zone MIB

- jnxLsysSpZoneMIB on page 438
- jnxLsysSpZoneObjects on page 439
- jnxLsysSpZoneSummary on page 440

jnxLsysSpZoneMIB

The Juniper Networks enterprise-specific zone MIB, whose object identifier is **{jnxLsysSpZone 1 }**, defines the zone-specific MIB for logical system security profiles.

Copyright © 2012, Juniper Networks, Inc.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-zone.txt .

For information about the enterprise-specific logical system zone-specific MIB objects, see the following topics:

- jnxLsysSpZoneObjects on page 439
- jnxLsysSpZoneSummary on page 440

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpZoneObjects

The **jnxLsysSpZoneObjects**, whose object identifier is **{jnxLsysSpZoneMIB 1}**, includes the logical system zone-specific objects as listed in Table 118 on page 439.

Table 118: jnxLsysSpZoneObjects

Object	Object Identifier	Description
jnxLsysSpZoneTable	**jnxLsysSpZoneObjects 1**	Represents the logical system profile zone objects for zone resource consumption per logical system.
jnxLsysSpZoneEntry	**jnxLsysSpZoneTable 1**	Represents an entry in the zone resource table.

Table 119 on page 439 lists and describes the objects contained in a **jnxLsysSpZoneEntry**.

Table 119: jnxLsysSpZoneEntry

Object	Object ID	Description
jnxLsysSpZoneLsysName	**jnxLsysSpZoneEntry 1**	Represents the name of the logical system that uses the zone resource information.
jnxLsysSpZoneProfileName	**jnxLsysSpZoneEntry 2**	Represents the security profile name string for the logical system.
jnxLsysSpZoneUsage	**jnxLsysSpZoneEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpZoneReserved	**jnxLsysSpZoneEntry 4**	Represents the reserved resource count for the logical system.
jnxLsysSpZoneMaximum	**jnxLsysSpZoneEntry 5**	Represents the maximum allowed resource usage count for the logical system.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- jnxLsysSpZoneMIB on page 438
- jnxLsysSpZoneSummary on page 440

jnxLsysSpZoneSummary

The **jnxLsysSpZoneSummary**, whose object identifier is **{jnxLsysSpZoneMIB 2 }**, represents the zone resource consumption summary.

Table 120 on page 440 lists and describes the zone resource consumption summary.

Table 120: jnxLsysSpZoneSummary

Object	Object ID	Description
jnxLsysSpZoneUsedAmount	jnxLsysSpZoneSummary 1	Represents the total zone resource consumption by the logical systems.
jnxLsysSpZoneMaxQuota	jnxLsysSpZoneSummary 2	Represents the maximum allocated zone resources for the whole device for all logical systems.
jnxLsysSpZoneAvailableAmount	jnxLsysSpZoneSummary 3	Represents the zone resources available in the whole device.
jnxLsysSpZoneHeaviestUsage	jnxLsysSpZoneSummary 4	Represents the maximum zone resource consumption by a logical system.
jnxLsysSpZoneHeaviestUser	jnxLsysSpZoneSummary 5	Represents the name of the logical system that consumes the maximum zone resources.
jnxLsysSpZoneLightestUsage	jnxLsysSpZoneSummary 6	Represents the minimum zone resource consumption by a logical system.
jnxLsysSpZoneLightestUser	jnxLsysSpZoneSummary 7	Represents name of the logical system that consumes the minimum zone resources.

Related Documentation

- jnxLsysSpZoneMIB on page 438
- jnxLsysSpZoneObjects on page 439

LSYS Scheduler MIB

- jnxLsysSpSchedulerMIB on page 441
- jnxLsysSpSchedulerObjects on page 441
- jnxLsysSpSchedulerSummary on page 442

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpSchedulerMIB

The Juniper Networks enterprise-specific scheduler MIB, whose object identifier is **{jnxLsysSpScheduler 1}**, defines the scheduler-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-scheduler.txt .

For information about the enterprise-specific logical system scheduler-specific MIB objects, see the following topics:

- jnxLsysSpSchedulerObjects on page 441
- jnxLsysSpSchedulerSummary on page 442

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpSchedulerObjects

The **jnxLsysSpSchedulerObjects**, whose object identifier is **{jnxLsysSpSchedulerMIB 1}**, includes the logical system scheduler-specific objects as listed in Table 121 on page 441:

Table 121: jnxLsysSpSchedulerObjects

Object	Object Identifier	Description
jnxLsysSpSchedulerTable	**jnxLsysSpSchedulerObjects 1**	Represents the logical system profile for scheduler objects for scheduler resource consumption per logical system.
jnxLsysSpSchedulerEntry	**jnxLsysSpSchedulerTable 1**	Identifies an entry in the scheduler resource table.

Table 122 on page 441 lists and describes the objects contained in a **jnxLsysSpSchedulerEntry**.

Table 122: jnxLsysSpSchedulerEntry

Object	Object ID	Description
jnxLsysSpSchedulerLsysName	**jnxLsysSpSchedulerEntry 1**	Represents the name of the logical system that uses the scheduler resource information.
jnxLsysSpSchedulerProfileName	**jnxLsysSpSchedulerEntry 2**	Represents the security profile name string for the logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 122: jnxLsysSpSchedulerEntry *(continued)*

Object	Object ID	Description
jnxLsysSpSchedulerUsage	**jnxLsysSpSchedulerEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpSchedulerReserved	**jnxLsysSpSchedulerEntry 4**	Represents the reserved resource count for the logical system.
jnxLsysSpSchedulerMaximum	**jnxLsysSpSchedulerEntry 5**	Represents the maximum allowed resource usage count for the logical system.

Related Documentation

- jnxLsysSpSchedulerMIB on page 441
- jnxLsysSpSchedulerSummary on page 442

jnxLsysSpSchedulerSummary

The **jnxLsysSpSchedulerSummary**, whose object identifier is **{jnxLsysSpSchedulerMIB 2}**, represents the scheduler resource consumption summary.

Table 123 on page 442 lists and describes the scheduler resource consumption summary.

Table 123: jnxLsysSpSchedulerSummary

Object	Object ID	Description
jnxLsysSpSchedulerUsedAmount	**jnxLsysSpSchedulerSummary 1**	Represents the total scheduler resource consumption by the logical systems.
jnxLsysSpSchedulerMaxQuota	**jnxLsysSpSchedulerSummary 2**	Represents the maximum allocated scheduler resources for the whole device for all logical systems.
jnxLsysSpSchedulerAvailableAmount	**jnxLsysSpSchedulerSummary 3**	Represents the scheduler resources available in the whole device.
jnxLsysSpSchedulerHeaviestUsage	**jnxLsysSpSchedulerSummary 4**	Represents the maximum scheduler resource consumption by a logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 123: jnxLsysSpSchedulerSummary *(continued)*

Object	Object ID	Description
jnxLsysSpSchedulerHeaviestUser	jnxLsysSpSchedulerSummary 5	Represents the name of the logical system that consumes the maximum scheduler resources.
jnxLsysSpSchedulerLightestUsage	jnxLsysSpSchedulerSummary 6	Represents the minimum scheduler resource consumption by a logical system.
jnxLsysSpSchedulerLightestUser	jnxLsysSpSchedulerSummary 7	Represents the name of the logical system that consumes the minimum scheduler resources.

Related Documentation

- jnxLsysSpSchedulerMIB on page 441
- jnxLsysSpSchedulerObjects on page 441

LSYS Policy MIB

- jnxLsysSpPolicyMIB on page 443
- jnxLsysSpPolicyObjects on page 444
- jnxLsysSpPolicySummary on page 444

jnxLsysSpPolicyMIB

The Juniper Networks enterprise-specific Policy objects MIB, whose object identifier is **{ jnxLsysSpPolicy 1 }**, defines the policy-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-policy.txt .

For information about the enterprise-specific logical system policy-specific MIB objects, see the following topics:

- jnxLsysSpPolicyObjects on page 444
- jnxLsysSpPolicySummary on page 444

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpPolicyObjects

The **jnxLsysSpPolicyObjects**, whose object identifier is **{ jnxLsysSpPolicyMIB 1}**, represents the logical system policy objects as listed in Table 124 on page 444.

Table 124: jnxLsysSpZoneObjects

Object	Object Identifier	Description
jnxLsysSpPolicyTable	**jnxLsysSpPolicyObjects 1**	Represents the logical system profile for policy resource consumption per logical system.
jnxLsysSpPolicyEntry	**jjnxLsysSpPolicyTable 1**	Identifies an entry in the policy resource table.

Table 125 on page 444 lists and describes the objects contained in a **{jnxLsysSpPolicyEntry}**.

Table 125: jnxLsysSpPolicyEntry

Object	Object ID	Description
jnxLsysSpPolicyLsysName	**jnxLsysSpPolicyEntry 1**	Represents the name of the logical system that uses the policy resource information.
jnxLsysSpPolicyProfileName	**jnxLsysSpPolicyEntry 2**	Represents the security profile name string for the logical system.
jnxLsysSpPolicyUsage	**jnxLsysSpPolicyEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpPolicyReserved	**jnxLsysSpPolicyEntry 4**	Represents the reserved resource count for the logical system.
jnxLsysSpPolicyMaximum	**jnxLsysSpPolicyEntry 5**	Represents the maximum allowed resource usage count for the logical system.

Related Documentation

- jnxLsysSpPolicyMIB on page 443
- jnxLsysSpPolicySummary on page 444

jnxLsysSpPolicySummary

The **jnxLsysSpPolicySummary** , whose object identifier is **{jnxLsysSpPolicyMIB 2}**, represents the policy resource consumption summary.

Table 126 on page 445 lists and describes the policy resource consumption summary.

Copyright © 2012, Juniper Networks, Inc.

Table 126: jnxLsysSpPolicySummary

Object	Object ID	Description
jnxLsysSpPolicyUsedAmount	jnxLsysSpPolicySummary 1	Represents the total policy resource consumption for all logical systems.
jnxLsysSpPolicyMaxQuota	jnxLsysSpPolicySummary 2	Represents the maximum allocated policy resources for the whole device for all logical systems on the device.
jnxLsysSpPolicyAvailableAmount	jnxLsysSpPolicySummary 3	Represents the amount of policy resources available for all logical systems on the device.
jnxLsysSpPolicyHeaviestUsage	jnxLsysSpPolicySummary 4	Represents the maximum policy resource consumption by a logical system.
jnxLsysSpPolicyHeaviestUser	jnxLsysSpPolicySummary 5	Represents the name of the logical system that consumes the maximum policy resources.
jnxLsysSpPolicyLightestUsage	jnxLsysSpPolicySummary 6	Represents the minimum policy resource consumption by a logical system.
jnxLsysSpPolicyLightestUser	jnxLsysSpPolicySummary 7	Represents the name of the logical system that consumes minimum policy resources.

Related Documentation

- jnxLsysSpPolicyMIB on page 443
- jnxLsysSpPolicyObjects on page 444

LSYS Policy with Count MIB

- jnxLsysSpPolicywcntMIB on page 445
- jnxLsysSpPolicywcntObjects on page 446
- jnxLsysSpPolicywcntSummary on page 447

jnxLsysSpPolicywcntMIB

The Juniper Networks enterprise-specific policy with count MIB, whose object identifier is **{jnxLsysSpPolicywcnt 1 }**, defines the policy-with-count-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-policywcnt.txt .

For information about the enterprise-specific logical system policy- with-count- specific MIB objects, see the following topics:

- jnxLsysSpPolicywcntObjects on page 446
- jnxLsysSpPolicywcntSummary on page 447

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpPolicywcntObjects

The **jnxLsysSpPolicywcntObjects**, whose object identifier is **{jnxLsysSpPolicywcntMIB 1}**, represents the logical systems policy-with-count specific objects as listed in Table 127 on page 446.

Table 127: jnxLsysSpPolicywcntObjects

Object	Object Identifier	Description
jnxLsysSpPolicywcntTable	**jnxLsysSpPolicywcntObjects 1**	Represents the logical system profile for policy-with-count resource consumption per logical system.
JnxLsysSpPolicywcntEntry	**jnxLsysSpPolicywcntTable 1**	Identifies an entry in the policy-with-count resource table.

Table 128 on page 446 lists and describes the objects contained in a **JnxLsysSpPolicywcntEntry**.

Table 128: JnxLsysSpPolicywcntEntry

Object	Object ID	Description
jnxLsysSpPolicywcntLsysName	**jnxLsysSpPolicywcntEntry 1**	Represents the name of the logical system that uses the policy-with-count resource information.
jnxLsysSpPolicywcntProfileName	**jnxLsysSpPolicywcntEntry 2**	Represents the security profile name string for the logical system.
jnxLsysSpPolicywcntUsage	**jnxLsysSpPolicywcntEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpPolicywcntReserved	**jnxLsysSpPolicywcntEntry 4**	Represents the reserved resource count for the logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 128: JnxLsysSpPolicywcntEntry *(continued)*

Object	Object ID	Description
jnxLsysSpPolicywcntMaximum	**jnxLsysSpPolicywcntEntry 5**	Represents the maximum allowed resource usage count for the logical system.

Related Documentation

- jnxLsysSpPolicywcntMIB on page 445
- jnxLsysSpPolicywcntSummary on page 447

jnxLsysSpPolicywcntSummary

The **jnxLsysSpPolicywcntSummary**, whose object identifier is **{jnxLsysSpPolicywcntMIB 2 }**, represents the policy-with-count resource consumption summary.

Table 129 on page 447 lists and describes the objects for policy-with-count resource consumption summary.

Table 129: jnxLsysSpPolicywcntMIB 2

Object	Object ID	Description
jnxLsysSpPolicywcntUsedAmount	**jnxLsysSpPolicywcntSummary 1**	Represents the total policy-with-count resource consumption by the logical systems.
jnxLsysSpPolicywcntMaxQuota	**jnxLsysSpPolicywcntSummary 2**	Represents the maximum allocated policy-with-count resource for the whole device for all logical systems.
jnxLsysSpPolicywcntAvailableAmount	**jnxLsysSpPolicywcntSummary 3**	Represents the policy-with-count resources available in the whole device.
jnxLsysSpPolicywcntHeaviestUsage	**jnxLsysSpPolicywcntSummary 4**	Represents the maximum amount of the policy-with-count resource consumption by a logical system.
jnxLsysSpPolicywcntHeaviestUser	**jnxLsysSpPolicywcntSummary 5**	Represents the name of the logical system that consumes the maximum amount of the policy-with-count resources.

Copyright © 2012, Juniper Networks, Inc.

Table 129: jnxLsysSpPolicywcntMIB 2 *(continued)*

Object	Object ID	Description
jnxLsysSpPolicywcntLightestUsage	**jnxLsysSpPolicywcntSummary 6**	Represents the minimum policy-with-count resource consumption by a logical system.
jnxLsysSpPolicywcntLightestUser	**jnxLsysSpPolicywcntSummary 7**	Represents the name of the logical system that consumes the minimum policy-with-count resource.

Related Documentation

- jnxLsysSpPolicywcntMIB on page 445
- jnxLsysSpPolicywcntObjects on page 446

LSYS Flow Gate MIB

- jnxLsysSpFlowgateMIB on page 448
- jnxLsysSpFlowgateObjects on page 448
- jnxLsysSpFlowgateSummary on page 449

jnxLsysSpFlowgateMIB

The Juniper Networks enterprise-specific LSYS Flow gate MIB, whose object identifier is **{jnxLsysSpFlowgate 1}**, defines the flow gate-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-flowgate.txt.

For information about the enterprise-specific logical system NAT flow- gate-specific MIB objects, see the following topics:

- jnxLsysSpFlowgateObjects on page 448
- jnxLsysSpFlowgateSummary on page 449

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpFlowgateObjects

The **jnxLsysSpFlowgateObjects**, whose object identifier is **{jnxLsysSpFlowgateMIB 1 }**, represents the logical system flow- gate-specific objects as listed in Table 130 on page 449.

 Copyright © 2012, Juniper Networks, Inc.

Table 130: jnxLsysSpFlowgateObjects

Object	Object ID	Description
jnxLsysSpFlowgateTable	**jnxLsysSpFlowgateObjects 1**	Represents the logical system profile flow gate objects for flow gate resource consumption per logical system.
jnxLsysSpFlowgateEntry	**jnxLsysSpFlowgateTable 1**	Represents an entry in the flow gate resource table.

The **JnxLsysSpFlowgateEntry** includes the objects that are listed in Table 131 on page 449.

Table 131: JnxLsysSpFlowgateEntry

Object	Object ID	Description
jnxLsysSpFlowgateLsysName	**jnxLsysSpFlowgateEntry 1**	Represents the name of the logical system for which flow gate resource information is retrieved.
jnxLsysSpFlowgateProfileName	**jnxLsysSpFlowgateEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpFlowgateUsage	**jnxLsysSpFlowgateEntry 3**	Represents the current resource usage count for a logical system.
jnxLsysSpFlowgateReserved	**jnxLsysSpFlowgateEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpFlowgateMaximum	**jnxLsysSpFlowgateEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpFlowgateMIB on page 448
- jnxLsysSpFlowgateSummary on page 449

jnxLsysSpFlowgateSummary

The **jnxLsysSpFlowgateSummary**, whose object identifier is **{jnxLsysSpFlowgateMIB 2}**, represents the flow gate resource consumption summary.

Table 132 on page 449 lists and describes the flow-gate resource consumption summary.

Table 132: jnxLsysSpFlowgateSummary

Object	Object ID	Description
jnxLsysSpFlowgateUsedAmount	**jnxLsysSpFlowgateSummary 1**	Represents the flow gate resource consumption for all logical systems.
jnxLsysSpFlowgateMaxQuota	**jnxLsysSpFlowgateSummary 2**	Represents the flow gate resource maximum quota for all logical systems on the device.

Copyright © 2012, Juniper Networks, Inc.

Table 132: jnxLsysSpFlowgateSummary *(continued)*

Object	Object ID	Description
jnxLsysSpFlowgateAvailableAmount	**jnxLsysSpFlowgateSummary 3**	Represents the amount of flow gate resources available for all logical systems on the device.
jnxLsysSpFlowgateHeaviestUsage	**jnxLsysSpFlowgateSummary 4**	Represents the most used flow gate resources consumed by a logical system.
jnxLsysSpFlowgateHeaviestUser	**jnxLsysSpFlowgateSummary 5**	Represents the logical system name that consumes the most flow-gate resources.
jnxLsysSpFlowgateLightestUsage	**jnxLsysSpFlowgateSummary 6**	Represents the least used flow-gate resources consumed by a logical system.
jnxLsysSpFlowgateLightestUser	**jnxLsysSpFlowgateSummary 7**	Represents the logical system name that consumes the least flow-gate resources.

Related Documentation

- jnxLsysSpFlowgateMIB on page 448
- jnxLsysSpFlowgateObjects on page 448

LSYS Flow Session MIB

- jnxLsysSpFlowsessMIB on page 450
- jnxLsysSpFlowsessObjects on page 451
- jnxLsysSpFlowsessSummary on page 451

jnxLsysSpFlowsessMIB

The Juniper Networks enterprise-specific LSYS flow session MIB, whose object identifier is **{jnxLsysSpFlowsess 1}**, defines the flow session-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-flowsess.txt.

For information about the enterprise-specific logical system flow session MIB objects, see the following topics:

- jnxLsysSpFlowsessObjects on page 451
- jnxLsysSpFlowsessSummary on page 451

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpFlowsessObjects

The **jnxLsysSpFlowsessObjects**, whose object identifier is **{jnxLsysSpFlowsessMIB 1 }**, represents the logical system flow session specific objects as listed in Table 133 on page 451.

Table 133: jnxLsysSpFlowsessObjects

Object	Object ID	Description
jnxLsysSpFlowsessTable	**jnxLsysSpFlowsessObjects 1**	Represents the logical system profile flow session objects for flow session resource consumption per logical system.
jnxLsysSpFlowsessEntry	**jnxLsysSpFlowsessTable 1**	Represents an entry in the flow session resource table.

The **JnxLsysSpFlowsessEntry** includes the objects that are listed in Table 134 on page 451.

Table 134: JnxLsysSpFlowsessEntry

Object	Object ID	Description
jnxLsysSpFlowsessLsysName	**jnxLsysSpFlowsessEntry 1**	Represents the name of the logical system for which flow session resource information is retrieved.
jnxLsysSpFlowsessProfileName	**jnxLsysSpFlowsessEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpFlowsessUsage	**jnxLsysSpFlowsessEntry 3**	Represents the current resource usage count for a logical system.
jnxLsysSpFlowsessReserved	**jnxLsysSpFlowsessEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpFlowsessMaximum	**jnxLsysSpFlowsessEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpFlowsessMIB on page 450
- jnxLsysSpFlowsessSummary on page 451

jnxLsysSpFlowsessSummary

The **jnxLsysSpFlowsessSummary**, whose object identifier is **{jnxLsysSpFlowsessMIB 2}**, represents the flow session resource consumption summary.

Table 135 on page 452 lists and describes the flow session resource consumption summary.

Table 135: jnxLsysSpFlowsessSummary

Object	Object ID	Description
jnxLsysSpFlowsessUsedAmount	jnxLsysSpFlowsessSummary 1	Represents the flow session resource consumption for all logical systems.
jnxLsysSpFlowsessMaxQuota	jnxLsysSpFlowsessSummary 2	Represents the flow session resource maximum quota for all logical systems on the device.
jnxLsysSpFlowsessAvailableAmount	jnxLsysSpFlowsessSummary 3	Represents the amount of flow session resources available for all logical systems on the device.
jnxLsysSpFlowsessHeaviestUsage	jnxLsysSpFlowsessSummary 4	Represents the most used flow session resources consumed by a logical system.
jnxLsysSpFlowsessHeaviestUser	jnxLsysSpFlowsessSummary 5	Represents the logical system name that consumes the most flow session resources.
jnxLsysSpFlowsessLightestUsage	jnxLsysSpFlowsessSummary 6	Represents the least used flow session resources consumed by a logical system.
jnxLsysSpFlowsessLightestUser	jnxLsysSpFlowsessSummary 7	Represents the logical system name that consumes the least flow session resources.

Related Documentation

- jnxLsysSpFlowsessMIB on page 450
- jnxLsysSpFlowsessObjects on page 451

LSYS AUTH Entry MIB

- jnxLsysSpAuthentryMIB on page 452
- jnxLsysSpAuthentryObjects on page 453
- jnxLsysSpAuthentrySummary on page 453

jnxLsysSpAuthentryMIB

The Juniper Networks enterprise-specific LSYS NAT Auth entry MIB, whose object identifier is **{jnxLsysSpAuthentry 1}**, defines the authentication-entry-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-authentry.txt.

For information about the enterprise-specific logical system auth entry pool MIB objects, see the following topics:

- jnxLsysSpAuthentryObjects on page 453

Copyright © 2012, Juniper Networks, Inc.

- jnxLsysSpAuthentrySummary on page 453

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpAuthentryObjects

The **jnxLsysSpAuthentryObjects**, whose object identifier is **{jnxLsysSpAuthentryMIB 1}**, represents the logical system NAT authentication entry objects as listed in Table 136 on page 453.

Table 136: jnxLsysSpAuthentryObjects

Object	Object ID	Description
jnxLsysSpAuthentryTable	jnxLsysSpAuthentryObjects 1	Represents the logical system profile objects for authentication entry resource consumption per logical system.
jnxLsysSpAuthentryEntry	jnxLsysSpAuthentryTable 1	Represents an entry in the authentication entry resource table.

The **JnxLsysSpAuthentryEntry** includes the objects that are listed in Table 137 on page 453.

Table 137: jnxLsysSpAuthentryEntry

Object	Object ID	Description
jnxLsysSpAuthentryLsysName	jnxLsysSpAuthentryEntry 1	Represents the name of the logical system for which authentication entry resource information is retrieved.
jnxLsysSpAuthentryProfileName	jnxLsysSpAuthentryEntry 2	Represents the security profile name string for a logical system.
jnxLsysSpAuthentryUsage	jnxLsysSpAuthentryEntry 3	Represents the current resource usage count for a logical system.
jnxLsysSpAuthentryReserved	jnxLsysSpAuthentryEntry 4	Represents the reserved resource count for a logical system.
jnxLsysSpAuthentryMaximum	jnxLsysSpAuthentryEntry 5	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpAuthentryMIB on page 452
- jnxLsysSpAuthentrySummary on page 453

jnxLsysSpAuthentrySummary

The **jnxLsysSpAuthentrySummary**, whose object identifier is **{jnxLsysSpAuthentryMIB 2}**, represents the authentication entry resource consumption summary.

Table 138 on page 454 lists and describes the authentication entry resource consumption summary.

Table 138: jnxLsysSpAuthentrySummary

Object	Object ID	Description
jnxLsysSpAuthentryUsedAmount	jnxLsysSpAuthentrySummary 1	Represents the authentication entrye resource consumption for all logical systems.
jnxLsysSpAuthentryMaxQuota	jnxLsysSpAuthentrySummary 2	Represents the authentication entry resource maximum quota for all logical systems on the device.
jnxLsysSpAuthentryAvailableAmount	jnxLsysSpAuthentrySummary 3	Represents the amount of authentication entry resources available for all logical systems on the device.
jnxLsysSpAuthentryHeaviestUsage	jnxLsysSpAuthentrySummary 4	Represents the most used authentication entry resources consumed by a logical system.
jnxLsysSpAuthentryHeaviestUser	jnxLsysSpAuthentrySummary 5	Represents the logical system name that consumes the most authentication entry resources.
jnxLsysSpAuthentryLightestUsage	jnxLsysSpAuthentrySummary 6	Represents the least used authentication entry resources consumed by a logical system.
jnxLsysSpAuthentryLightestUser	jnxLsysSpAuthentrySummary 7	Represents the logical system name that consumes the least authentication entry resources.

Related Documentation

- jnxLsysSpAuthentryMIB on page 452
- jnxLsysSpAuthentryObjects on page 453

LSYS NAT Source Pool MIB

- jnxLsysSpNATsrcpoolMIB on page 454
- jnxLsysSpNATsrcpoolObjects on page 455
- jnxLsysSpNATsrcpoolSummary on page 456

jnxLsysSpNATsrcpoolMIB

The Juniper Networks enterprise-specific LSYS NAT Source Pool MIB, whose object identifier is **{jnxLsysSpNATsrcpool 1}**, defines the NAT source-pool-specific MIB for logical system security profiles.

Copyright © 2012, Juniper Networks, Inc.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natsrcpool.txt .

For information about the enterprise-specific logical system NAT source pool MIB objects, see the following topics:

- jnxLsysSpNATsrcpoolObjects on page 455
- jnxLsysSpNATsrcpoolSummary on page 456

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATsrcpoolObjects

The **jnxLsysSpNATsrcpoolObjects**, whose object identifier is **{jnxLsysSpNATsrcpoolMIB 1}**, represents the logical system NAT source pool objects as listed in Table 139 on page 455.

Table 139: jnxLsysSpNATsrcpoolObjects

Object	Object ID	Description
jnxLsysSpNATsrcpoolTable	**jnxLsysSpNATsrcpoolObjects 1**	Represents the logical system profile NAT source pool objects for NAT source pool resource consumption per logical system.
jnxLsysSpNATsrcpoolEntry	**jnxLsysSpNATsrcpoolTable 1**	Represents an entry in the NAT source pool resource table.

The **jnxLsysSpNATstaticruleEntry** includes the objects that are listed in Table 140 on page 455.

Table 140: jnxLsysSpNATsrcpoolEntry

Object	Object ID	Description
jnxLsysSpNATsrcpoolLsysName	**jnxLsysSpNATsrcpoolEntry 1**	Represents the name of the logical system for which NAT source pool resource information is retrieved.
jnxLsysSpNATsrcpoolProfileName	**jnxLsysSpNATsrcpoolEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpNATsrcpoolUsage	**jnxLsysSpNATsrcpoolEntry 3**	Represents the current resource usage count for a logical system.
jnxLsysSpNATsrcpoolReserved	**jnxLsysSpNATsrcpoolEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpNATsrcpoolMaximum	**jnxLsysSpNATsrcpoolEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- jnxLsysSpNATsrcpoolMIB on page 454
- jnxLsysSpNATsrcpoolSummary on page 456

jnxLsysSpNATsrcpoolSummary

The **jnxLsysSpNATsrcpoolSummary**, whose object identifier is **{jnxLsysSpNATsrcpoolMIB 2}**, represents the NAT source pool resource consumption summary.

Table 141 on page 456 lists and describes the NAT source pool resource consumption summary.

Table 141: jnxLsysSpNATsrcpoolSummary

Object	Object ID	Description
jnxLsysSpNATsrcpoolUsedAmount	jnxLsysSpNATsrcpoolSummary 1	Represents the NAT source pool resource consumption for all logical systems.
jnxLsysSpNATsrcpoolMaxQuota	jnxLsysSpNATsrcpoolSummary 2	Represents the NAT source pool resource maximum quota for all logical systems on the device.
jnxLsysSpNATsrcpoolAvailableAmount	jnxLsysSpNATsrcpoolSummary 3	Represents the amount of NAT source pool resources available for all logical systems on the device.
jnxLsysSpNATsrcpoolHeaviestUsage	jnxLsysSpNATsrcpoolSummary 4	Represents the most used NAT source pool resources consumed by a logical system.
jnxLsysSpNATsrcpoolHeaviestUser	jnxLsysSpNATsrcpoolSummary 5	Represents the logical system name that consumes the most NAT source pool resources.
jnxLsysSpNATsrcpoolLightestUsage	jnxLsysSpNATsrcpoolSummary 6	Represents the least used NAT source pool resources consumed by a logical system.
jnxLsysSpNATsrcpoolLightestUser	jnxLsysSpNATsrcpoolSummary 7	Represents the logical system name that consumes the least NAT source pool resources.

Related Documentation

- jnxLsysSpNATsrcpoolObjects on page 455
- jnxLsysSpNATsrcpoolMIB on page 454

LSYS NAT Destination Pool MIB

- jnxLsysSpNATdstpoolMIB on page 457
- jnxLsysSpNATdstpoolObjects on page 457
- jnxLsysSpNATdstpoolSummary on page 458

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpNATdstpoolMIB

The Juniper Networks enterprise-specific LSYS NAT Destination Pool MIB, whose object identifier is **{jnxLsysSpNATdstpool 1}**, defines the NAT destination-pool-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natsrcpool.txt .

For information about the enterprise-specific logical system NAT source pool MIB objects, see the following topics:

- jnxLsysSpNATdstpoolObjects on page 457
- jnxLsysSpNATdstpoolSummary on page 458

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATdstpoolObjects

The **jnxLsysSpNATdstpoolObjects**, whose object identifier is **{jnxLsysSpNATdstpoolMIB 1}**, as listed in Table 142 on page 457.

Table 142: jnxLsysSpNATdstpoolObjects

Object	Object ID	Description
jnxLsysSpNATdstpoolTable	**jnxLsysSpNATdstpoolObjects 1**	Represents the logical system profile NAT destination pool objects for NAT destination pool resource consumption per logical system.
jnxLsysSpNATdstpoolEntry	**jnxLsysSpNATdstpoolTable 1**	Represents an entry in the NAT destination pool resource table.

The **jnxLsysSpNATdstpoolEntry** includes the objects that are listed in Table 143 on page 457.

Table 143: jnxLsysSpNATdstpoolEntry

Object	Object ID	Description
jnxLsysSpNATdstpoolLsysName	**jnxLsysSpNATdstpoolEntry 1**	Represents the name of the logical system for which NAT destination pool resource information is retrieved.
jnxLsysSpNATdstpoolProfileName	**jnxLsysSpNATdstpoolEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpNATdstpoolUsage	**jnxLsysSpNATdstpoolEntry 3**	Represents the current resource usage count for a logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 143: jnxLsysSpNATdstpoolEntry *(continued)*

Object	Object ID	Description
jnxLsysSpNATdstpoolReserved	**jnxLsysSpNATdstpoolEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpNATdstpoolMaximum	**jnxLsysSpNATdstpoolEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATdstpoolMIB on page 457
- jnxLsysSpNATdstpoolSummary on page 458

jnxLsysSpNATdstpoolSummary

The **jnxLsysSpNATdstpoolSummary**, whose object identifier is **{jnxLsysSpNATdstpoolMIB 2}**, represents the NAT destination pool resource consumption summary.

Table 144 on page 458 lists and describes the NAT destination pool resource consumption summary.

Table 144: jnxLsysSpNATdstpoolSummary

Object	Object ID	Description
jnxLsysSpNATdstpoolUsedAmount	**jnxLsysSpNATdstpoolSummary 1**	Represents the NAT destination pool resource consumption for all logical systems.
jnxLsysSpNATdstpoolMaxQuota	**jnxLsysSpNATdstpoolSummary 2**	Represents the NAT destination pool resource maximum quota for all logical systems on the device.
jnxLsysSpNATdstpoolAvailableAmount	**jnxLsysSpNATdstpoolSummary 3**	Represents the amount of NAT destination pool resources available for all logical systems on the device.
jnxLsysSpNATdstpoolHeaviestUsage	**jnxLsysSpNATdstpoolSummary 4**	Represents the most used NAT destination pool resources consumed by a logical system.
jnxLsysSpNATdstpoolHeaviestUser	**jnxLsysSpNATdstpoolSummary 5**	Represents the logical system name that consumes the most NAT destination pool resources.
jnxLsysSpNATdstpoolLightestUsage	**jnxLsysSpNATdstpoolSummary 6**	Represents the least used NAT destination pool resources consumed by a logical system.
jnxLsysSpNATdstpoolLightestUser	**jnxLsysSpNATdstpoolSummary 7**	Represents the logical system name that consumes the least NAT destination pool resources.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- jnxLsysSpNATdstpoolMIB on page 457
- jnxLsysSpNATdstpoolObjects on page 457

LSYS NAT Source PAT Address MIB

- jnxLsysSpNATsrcpatadMIB on page 459
- jnxLsysSpNATsrcpatadObjects on page 459
- jnxLsysSpNATsrcpatadSummary on page 460

jnxLsysSpNATsrcpatadMIB

The Juniper Networks enterprise-specific LSYS NAT source PAT address MIB, whose object identifier is **{jnxLsysSpNATsrcpatad 1}**, defines the NAT source PAT address-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natsrcpatad.txt .

For information about the enterprise-specific logical system NAT source PAT address-specific MIB objects, see the following topics:

- jnxLsysSpNATsrcpatadObjects on page 459
- jnxLsysSpNATsrcpatadSummary on page 460

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATsrcpatadObjects

The **jnxLsysSpNATsrcpatadObjects**, whose object identifier is **{jnxLsysSpNATsrcpatadMIB 1}**, represents the logical system NAT source PAT address objects as listed in Table 145 on page 459.

Table 145: jnxLsysSpNATsrcpatadObjects

Object	Object ID	Description
jnxLsysSpNATsrcpatadTable	**jnxLsysSpNATsrcpatadObjects 1**	Represents the logical system profile NAT source PAT address objects for NAT source PAT address resource consumption per logical system.
jnxLsysSpNATsrcpatadEntry	**jnxLsysSpNATsrcpatadTable 1**	Represents an entry in the NAT source PAT address resource table.

The **jnxLsysSpNATsrcpatadEntry** includes the objects that are listed in Table 146 on page 460.

Table 146: jnxLsysSpNATsrcpatadEntry

Object	Object ID	Description
jnxLsysSpNATsrcpatadLsysName	jnxLsysSpNATsrcpatadEntry 1	Represents the name of the logical system for which NAT source PAT address resource information is retrieved.
jnxLsysSpNATsrcpatadProfileName	jnxLsysSpNATsrcpatadEntry 2	Represents the security profile name string for a logical system.
jnxLsysSpNATsrcpatadUsage	jnxLsysSpNATsrcpatadEntry 3	Represents the current resource usage count for a logical system.
jnxLsysSpNATsrcpatadReserved	jnxLsysSpNATsrcpatadEntry 4	Represents the reserved resource count for a logical system.
jnxLsysSpNATsrcpatadMaximum	jnxLsysSpNATsrcpatadEntry 5	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATsrcpatadMIB on page 459
- jnxLsysSpNATsrcpatadSummary on page 460

jnxLsysSpNATsrcpatadSummary

The **jnxLsysSpNATsrcpatadSummary**, whose object identifier is **{jnxLsysSpNATsrcpatadMIB 2}**, represents the represents the logical system NAT source PAT address resource consumption summary.

Table 147 on page 460 lists and describes NAT source PAT address resource consumption summary.

Table 147: jnxLsysSpNATsrcpatadSummary

Object	Object ID	Description
jnxLsysSpNATsrcpatadUsedAmount	jnxLsysSpNATsrcpatadSummary 1	Represents the NAT source PAT address resource consumption for all logical systems.
jnxLsysSpNATsrcpatadMaxQuota	jnxLsysSpNATsrcpatadSummary 2	Represents the NAT source PAT address resource maximum quota for all logical systems on the device.
jnxLsysSpNATsrcpatadAvailableAmount	jnxLsysSpNATsrcpatadSummary 3	Represents the amount of NAT source PAT address resources available for all logical systems on the device.
jnxLsysSpNATsrcpatadHeaviestUsage	jnxLsysSpNATsrcpatadSummary 4	Represents the most used NAT source PAT address resources consumed by a logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 147: jnxLsysSpNATsrcpatadSummary *(continued)*

Object	Object ID	Description
jnxLsysSpNATsrcpatadHeaviestUser	jnxLsysSpNATsrcpatadSummary 5	Represents the logical system name that consumes the most NAT source PAT address resources.
jnxLsysSpNATsrcpatadLightestUsage	jnxLsysSpNATsrcpatadSummary 6	Represents the least used NAT source PAT address resources consumed by a logical system.
jnxLsysSpNATsrcpatadLightestUser	jnxLsysSpNATsrcpatadSummary 7	Represents the logical system name that consumes the least NAT source PAT address resources.

Related Documentation

- jnxLsysSpNATsrcpatadMIB on page 459
- jnxLsysSpNATsrcpatadObjects on page 459

LSYS NAT Source No PAT Address MIB

- jnxLsysSpNATsrcnopatadMIB on page 461
- jnxLsysSpNATsrcnopatadObjects on page 462
- jnxLsysSpNATsrcnopatadSummary on page 462

jnxLsysSpNATsrcnopatadMIB

The Juniper Networks enterprise-specific LSYS NAT Destination Rule MIB, whose object identifier is **{jnxLsysSpNATsrcnopatad 1}**, defines the LSYS NAT source no PAT address-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natsrcnopatad.txt .

For information about the enterprise-specific LSYS NAT source no PAT address-specific MIB objects, see the following topics:

- jnxLsysSpNATsrcnopatadObjects on page 462
- jnxLsysSpNATsrcnopatadSummary on page 462

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpNATsrcnopatadObjects

The **jnxLsysSpNATsrcnopatadObjects**, whose object identifier is **{jnxLsysSpNATsrcnopatadMIB 1}**, represents the logical system NAT source no PAT address objects as listed in Table 148 on page 462.

Table 148: jnxLsysSpNATsrcnopatadObjects

Object	Object ID	Description
jnxLsysSpNATsrcnopatadTable	**jnxLsysSpNATsrcnopatadObjects 1**	Represents the logical system profile NAT source no PAT address objects for NAT source no PAT address resource consumption per logical system.
jnxLsysSpNATsrcnopatadEntry	**jnxLsysSpNATsrcnopatadTable 1**	Represents an entry in the NAT source no PAT address resource table.

The **jnxLsysSpNATsrcnopatadEntry** includes the objects that are listed in Table 149 on page 462.

Table 149: jnxLsysSpNATsrcnopatadEntry

Object	Object ID	Description
jnxLsysSpNATsrcnopatadLsysName	**jnxLsysSpNATsrcnopatadEntry 1**	Represents the name of the logical system for which NAT source no PAT address resource information is retrieved.
jnxLsysSpNATsrcnopatadProfileName	**jnxLsysSpNATsrcnopatadEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpNATsrcnopatadUsage	**jnxLsysSpNATsrcnopatadEntry 3**	Represents the current resource usage count for a logical system.
jnxLsysSpNATsrcnopatadReserved	**jnxLsysSpNATsrcnopatadEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpNATsrcnopatadMaximum	**jnxLsysSpNATsrcnopatadEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATsrcnopatadMIB on page 461
- jnxLsysSpNATsrcnopatadSummary on page 462

jnxLsysSpNATsrcnopatadSummary

The **jnxLsysSpNATsrcnopatadSummary**, whose object identifier is **{jnxLsysSpNATsrcnopatadMIB 2}**, represents the NAT source no PAT address resource consumption summary.

Copyright © 2012, Juniper Networks, Inc.

Table 150 on page 463 lists and describes the NAT source pool resource consumption summary.

Table 150: jnxLsysSpNATsrcnopatadSummary

Object	Object ID	Description
jnxLsysSpNATsrcnopatadUsedAmount	jnxLsysSpNATsrcnopatadSummary 1	Represents the NAT source no PAT address resource consumption for all logical systems.
jnxLsysSpNATsrcnopatadMaxQuota	jnxLsysSpNATsrcnopatadSummary 2	Represents the NAT source no PAT address resource maximum quota for all logical systems on the device.
jnxLsysSpNATsrcnopatadAvailableAmount	jnxLsysSpNATsrcnopatadSummary 3	Represents the amount of NAT source no PAT address resources available for all logical systems on the device.
jnxLsysSpNATsrcnopatadHeaviestUsage	jnxLsysSpNATsrcnopatadSummary 4	Represents the most used NAT source no PAT address resources consumed by a logical system.
jnxLsysSpNATsrcnopatadHeaviestUser	jnxLsysSpNATsrcnopatadSummary 5	Represents the logical system name that consumes the most NAT source no PAT address resources.
jnxLsysSpNATsrcnopatadLightestUsage	jnxLsysSpNATsrcnopatadSummary 6	Represents the least used NAT source no PAT address resources consumed by a logical system.
jnxLsysSpNATsrcnopatadLightestUser	jnxLsysSpNATsrcnopatadSummary 7	Represents the logical system name that consumes the least NAT source no PAT address resources.

Related Documentation

- jnxLsysSpNATsrcnopatadMIB on page 461
- jnxLsysSpNATsrcnopatadObjects on page 462

LSYS NAT Source Rule MIB

- jnxLsysSpNATsrcruleMIB on page 463
- jnxLsysSpNATsrcruleObjects on page 464
- jnxLsysSpNATsrcruleSummary on page 465

jnxLsysSpNATsrcruleMIB

The Juniper Networks enterprise-specific NAT source rule MIB, whose object identifier is {**jnxLsysSpNATsrcrule 1**}, defines the source-rule-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natsrcrule.txt .

For information about the enterprise-specific logical system NAT source rule-specific MIB objects, see the following topics:

- jnxLsysSpNATsrcruleObjects on page 464
- jnxLsysSpNATsrcruleSummary on page 465

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATsrcruleObjects

The **jnxLsysSpNATsrcruleObjects**, whose object identifier is { **jnxLsysSpNATsrcruleMIB 1** }, represents the logical system NAT source rule objects as listed in Table 151 on page 464.

Table 151: jnxLsysSpNATsrcruleObjects

Object	Object Identifier	Description
jnxLsysSpNATsrcruleTable	**jnxLsysSpNATsrcruleObjects 1**	Represents the logical system profile for NAT source rule resource consumption per logical system.
jnxLsysSpNATsrcruleEntry	**jnxLsysSpNATsrcruleTable 1**	Identifies an entry in the NAT source rule resource table.

The **jnxLsysSpNATsrcruleEntry** includes the objects that are listed in Table 152 on page 464.

Table 152: jnxLsysSpNATsrcruleEntry

Object	Object ID	Description
jnxLsysSpNATsrcruleLsysName	**jnxLsysSpNATsrcruleEntry 1**	Represents the name of the logical system that uses the NAT source rule resource information.
jnxLsysSpNATsrcruleProfileName	**jnxLsysSpNATsrcruleEntry 2**	Represents the security profile name string for the logical system.
jnxLsysSpNATsrcruleUsage	**jnxLsysSpNATsrcruleEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpNATsrcruleReserved	**jnxLsysSpNATsrcruleEntry 4**	Represents the reserved resource count for the logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 152: jnxLsysSpNATsrcruleEntry *(continued)*

Object	Object ID	Description
jnxLsysSpNATsrcruleMaximum	jnxLsysSpNATsrcruleEntry 5	Represents the maximum allowed resource usage count for the logical system.

Related Documentation

- jnxLsysSpNATsrcruleMIB on page 463
- jnxLsysSpNATsrcruleSummary on page 465

jnxLsysSpNATsrcruleSummary

The **jnxLsysSpNATsrcruleSummary** includes the objects that are listed in Table 153 on page 465.

Table 153: jnxLsysSpNATsrcruleSummary

Object	Object ID	Description
jnxLsysSpNATsrcruleUsedAmount	jnxLsysSpNATsrcruleSummary 1	Represents the total amount of NAT source rule resource consumption by the logical systems.
jnxLsysSpNATsrcruleMaxQuota	jnxLsysSpNATsrcruleSummary 2	Represents the maximum allocated NAT source rule resource for the whole device for all logical systems.
jnxLsysSpNATsrcruleAvailableAmount	jnxLsysSpNATsrcruleSummary 3	Represents the NAT source rule resource available in the whole device.
jnxLsysSpNATsrcruleHeaviestUsage	jnxLsysSpNATsrcruleSummary 4	Represents the maximum NAT source rule resource consumption by a logical system.
jnxLsysSpNATsrcruleHeaviestUser	jnxLsysSpNATsrcruleSummary 5	Represents the name of the logical system that consumes the most NAT source rule resources.

Copyright © 2012, Juniper Networks, Inc.

Table 153: jnxLsysSpNATsrcruleSummary *(continued)*

Object	Object ID	Description
jnxLsysSpNATsrcruleLightestUsage	**jnxLsysSpNATsrcruleSummary 6**	Represents the minimum NAT source rule resource consumption by a logical system.
jnxLsysSpNATsrcruleLightestUser	**jnxLsysSpNATsrcruleSummary 7**	Represents the name of the logical system that consumes the least NAT source rule resources.

Related Documentation

- jnxLsysSpNATsrcruleMIB on page 463
- jnxLsysSpNATsrcruleObjects on page 464

LSYS NAT Destination Rule MIB

- jnxLsysSpNATdstruleMIB on page 466
- jnxLsysSpNATdstruleObjects on page 467
- jnxLsysSpNATdstruleSummary on page 467

jnxLsysSpNATdstruleMIB

The Juniper Networks enterprise-specific LSYS NAT Destination Rule MIB, whose object identifier is **{jnxLsysSpNATdstrule 1}**, defines the NAT destination-rule-specific MIB for logical system security profiles.

The logical systems security profile provides various type of static and dynamic resource management by observing resource allocation limits.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natdstrule.txt .

For information about the enterprise-specific LSYS NAT destination-rule-specific MIB objects, see the following topics:

- jnxLsysSpNATdstruleObjects on page 467
- jnxLsysSpNATdstruleSummary on page 467

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpNATdstruleObjects

The **jnxLsysSpNATdstruleObjects**, whose object identifier is **{jnxLsysSpNATdstruleMIB 1}**, represents the logical system NAT destination rule objects as listed in Table 154 on page 467.

Table 154: jnxLsysSpNATdstruleObjects

Object	Object ID	Description
jnxLsysSpNATdstruleTable	jnxLsysSpNATdstruleObjects 1	Represents the logical system profile NAT destination rule objects for NAT destination pool resource consumption per logical system.
jnxLsysSpNATdstruleEntry	jnxLsysSpNATdstruleTable 1	Represents an entry in the NAT destination rule resource table.

The **jnxLsysSpNATdstruleEntry** includes the objects that are listed in Table 155 on page 467.

Table 155: jnxLsysSpNATdstruleEntry

Object	Object ID	Description
jnxLsysSpNATdstruleLsysName	jnxLsysSpNATdstruleEntry 1	Represents the name of the logical system for which NAT destination rule resource information is retrieved.
jnxLsysSpNATdstruleProfileName	jnxLsysSpNATdstruleEntry 2	Represents the security profile name string for a logical system.
jnxLsysSpNATdstruleUsage	jnxLsysSpNATdstruleEntry 3	Represents the current resource usage count for a logical system.
jnxLsysSpNATdstruleReserved	jnxLsysSpNATdstruleEntry 4	Represents the reserved resource count for a logical system.
jnxLsysSpNATdstruleMaximum	jnxLsysSpNATdstruleEntry 5	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATdstruleMIB on page 466
- jnxLsysSpNATdstruleSummary on page 467

jnxLsysSpNATdstruleSummary

The **jnxLsysSpNATdstruleSummary**, whose object identifier is **{jnxLsysSpNATdstruleMIB 2}**, represents the NAT destination rule resource consumption summary.

Table 156 on page 468 lists and describes the NAT destination rule resource consumption summary.

Table 156: jnxLsysSpNATdstruleSummary

Object	Object ID	Description
jnxLsysSpNATdstruleUsedAmount	jnxLsysSpNATdstruleSummary 1	Represents the NAT destination rule resource consumption for all logical systems.
jnxLsysSpNATdstruleMaxQuota	jnxLsysSpNATdstruleSummary 2	Represents the NAT destination rule resource maximum quota for all logical systems on the device.
jnxLsysSpNATdstruleAvailableAmount	jnxLsysSpNATdstruleSummary 3	Represents the amount of NAT destination rule resources available for all logical systems on the device.
jnxLsysSpNATdstruleHeaviestUsage	jnxLsysSpNATdstruleSummary 4	Represents the most used NAT destination rule resources consumed by a logical system.
jnxLsysSpNATdstruleHeaviestUser	jnxLsysSpNATdstruleSummary 5	Represents the logical system name that consumes the most NAT source pool resources.
jnxLsysSpNATdstruleLightestUsage	jnxLsysSpNATdstruleSummary 6	Represents the least used NAT destination rule resources consumed by a logical system.
jnxLsysSpNATdstruleLightestUser	jnxLsysSpNATdstruleSummary 7	Represents the logical system name that consumes the least NAT destination rule resources.

Related Documentation

- jnxLsysSpNATdstruleMIB on page 466
- jnxLsysSpNATdstruleObjects on page 467

LSYS NAT Static Rule MIB

- jnxLsysSpNATstaticruleMIB on page 468
- jnxLsysSpNATstaticruleObjects on page 469
- jnxLsysSpNATstaticruleSummary on page 470

jnxLsysSpNATstaticruleMIB

The Juniper Networks enterprise-specific NAT static rule MIB, whose object identifier **{ jnxLsysSpNATstaticrule 1 }**, defines the static-rule-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natstaticrule.txt .

Copyright © 2012, Juniper Networks, Inc.

For information about the enterprise-specific logical system NAT static rule-specific MIB objects, see the following topics:

- jnxLsysSpNATstaticruleObjects on page 469
- jnxLsysSpNATstaticruleSummary on page 470

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATstaticruleObjects

The **jnxLsysSpNATstaticruleObjects**, whose object identifier is **{jnxLsysSpNATstaticruleMIB 1}**, represents the logical system NAT static rule objects as listed in Table 157 on page 469.

Table 157: jnxLsysSpNATstaticruleObjects

Object	Object Identifier	Description
jnxLsysSpNATstaticruleTable	**jnxLsysSpNATstaticruleObjects 1**	Represents the logical system profile for NAT static rule resource consumption per logical system.
jnxLsysSpNATstaticruleEntry	**jnxLsysSpNATstaticruleTable 1**	Identifies an entry in the NAT static rule resource table.

The **jnxLsysSpNATstaticruleEntry** includes the objects that are listed in Table 158 on page 469.

Table 158: jnxLsysSpNATstaticruleEntry

Object	Object ID	Description
jnxLsysSpNATstaticruleLsysName	**jnxLsysSpNATstaticruleEntry 1**	Represents the name of the logical system that uses the NAT static rule resource information.
jnxLsysSpNATstaticruleProfileName	**jnxLsysSpNATstaticruleEntry 2**	Represents the security profile name string for the logical system.
jnxLsysSpNATstaticruleUsage	**jnxLsysSpNATstaticruleEntry 3**	Represents the current resource usage count for the logical system.
jnxLsysSpNATstaticruleReserved	**jnxLsysSpNATstaticruleEntry 4**	Represents the reserved resource count for the logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 158: jnxLsysSpNATstaticruleEntry *(continued)*

Object	Object ID	Description
jnxLsysSpNATstaticruleMaximum	jnxLsysSpNATstaticruleEntry 5	Represents the maximum allowed resource usage count for the logical system.

Related Documentation

- jnxLsysSpNATstaticruleMIB on page 468
- jnxLsysSpNATstaticruleSummary on page 470

jnxLsysSpNATstaticruleSummary

The **jnxLsysSpNATstaticruleSummary** , whose object identifier is {**jnxLsysSpNATstaticruleMIB 2** }, defines the objects that are used to monitor NAT attributes as listed in Table 159 on page 470.

Table 159: jnxLsysSpNATstaticruleSummary

Object	Object ID	Description
jnxLsysSpNATstaticruleUsedAmount	jnxLsysSpNATstaticruleSummary1	Represents the total NAT static rule resource consumption by the logical systems.
jnxLsysSpNATstaticruleMaxQuota	jnxLsysSpNATstaticruleSummary 2	Represents the maximum allocated NAT static rule resources for the whole device for all logical systems.
jnxLsysSpNATstaticruleAvailableAmount	jnxLsysSpNATstaticruleSummary 3	Represents the NAT static rule resources available in the whole device.
jnxLsysSpNATstaticruleHeaviestUsage	jnxLsysSpNATstaticruleSummary 4	Represents the maximum NAT static rule resource consumption by a logical system.
jnxLsysSpNATstaticruleHeaviestUser	jnxLsysSpNATstaticruleSummary 5	Represents the name of the logical system that consumes the maximum NAT static rule resources.
jnxLsysSpNATstaticruleLightestUsage	jnxLsysSpNATstaticruleSummary 6	Represents the minimum NAT static rule resource consumption by a logical system.

Copyright © 2012, Juniper Networks, Inc.

Table 159: jnxLsysSpNATstaticruleSummary *(continued)*

Object	Object ID	Description
jnxlogical systemsSpNATsrcruleLightestUser	**jnxLsysSpNATstaticruleSummary 7**	Represents the name of the logical system that consumes the minimum NAT static rule resources.

Related Documentation

- jnxLsysSpNATstaticruleMIB on page 468
- jnxLsysSpNATstaticruleObjects on page 469

LSYS NAT Cone Bind MIB

- jnxLsysSpNATconebindMIB on page 471
- jnxLsysSpNATconebindObjects on page 471
- jnxLsysSpNATconebindSummary on page 472

jnxLsysSpNATconebindMIB

The Juniper Networks enterprise-specific LSYS NAT cone bind MIB, whose object identifier is **{jnxLsysSpNATconebind 1}**, defines the NAT cone bind-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natconebind.txt.

For information about the enterprise-specific logical system NAT cone bind MIB objects, see the following topics:

- jnxLsysSpNATconebindObjects on page 471
- jnxLsysSpNATconebindSummary on page 472

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpNATconebindObjects

The **jnxLsysSpNATconebindObjects**, whose object identifier is **{jnxLsysSpNATconebindMIB 1}**, represents the logical system NAT cone bind specific objects as listed in Table 160 on page 471.

Table 160: jnxLsysSpNATconebindObjects

Object	Object ID	Description
jnxLsysSpNATconebindTable	**jnxLsysSpNATconebindObjects 1**	Represents the logical system NAT cone bind objects for NAT cone bind resource consumption per logical system.

Table 160: jnxLsysSpNATconebindObjects *(continued)*

Object	Object ID	Description
jnxLsysSpNATconebindEntry	jnxLsysSpNATconebindTable 1	Represents an entry in the NAT cone bind resource table.

The **JnxLsysSpNATconebindEntry** includes the objects that are listed in Table 161 on page 472.

Table 161: JnxLsysSpNATconebindEntry

Object	Object ID	Description
jnxLsysSpNATconebindLsysName	jnxLsysSpNATconebindEntry 1	Represents the name of the logical system for which NAT cone bind resource information is retrieved.
jnxLsysSpNATconebindProfileName	jnxLsysSpNATconebindEntry 2	Represents the security profile name string for a logical system.
jnxLsysSpNATconebindUsage	jnxLsysSpNATconebindEntry 3	Represents the current resource usage count for a logical system.
jnxLsysSpNATconebindReserved	jnxLsysSpNATconebindEntry 4	Represents the reserved resource count for a logical system.
jnxLsysSpNATconebindMaximum	jnxLsysSpNATconebindEntry 5	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATconebindMIB on page 471
- jnxLsysSpNATconebindSummary on page 472

jnxLsysSpNATconebindSummary

The **jnxLsysSpNATconebindSummary**, whose object identifier is **{jnxLsysSpNATconebindMIB 2}**, represents the NAT cone bind resource consumption summary.

Table 162 on page 472 lists and describes the NAT cone bind resource consumption summary.

Table 162: jnxLsysSpNATconebindSummary

Object	Object ID	Description
jnxLsysSpNATconebindUsedAmount	jnxLsysSpNATconebindSummary 1	Represents the NAT cone bind resource consumption for all logical systems.
jnxLsysSpNATconebindMaxQuota	jnxLsysSpNATconebindSummary 2	Represents the NAT cone bind resource maximum quota for all logical systems on the device.

Copyright © 2012, Juniper Networks, Inc.

Table 162: jnxLsysSpNATconebindSummary *(continued)*

Object	Object ID	Description
jnxLsysSpNATconebindAvailableAmount	jnxLsysSpNATconebindSummary 3	Represents the amount of NAT cone bind resources available for all logical systems on the device.
jnxLsysSpNATconebindHeaviestUsage	jnxLsysSpNATconebindSummary 4	Represents the most used NAT cone bind resources consumed by a logical system.
jnxLsysSpNATconebindHeaviestUser	jnxLsysSpNATconebindSummary 5	Represents the logical system name that consumes the most NAT cone bind resources.
jnxLsysSpNATconebindLightestUsage	jnxLsysSpNATconebindSummary 6	Represents the least used NAT cone bind resources consumed by a logical system.
jnxLsysSpNATconebindLightestUser	jnxLsysSpNATconebindSummary 7	Represents the logical system name that consumes the least NAT cone bind resources.

Related Documentation

- jnxLsysSpNATconebindMIB on page 471
- jnxLsysSpNATconebindObjects on page 471

LSYS NAT Port Overloading IP Number MIB

- jnxLsysSpNATpoipnumMIB on page 473
- jnxLsysSpNATpoipnumObjects on page 474
- jnxLsysSpNATpoipnumSummary on page 474

jnxLsysSpNATpoipnumMIB

The Juniper Networks enterprise-specific LSYS NAT Source Pool MIB, whose object identifier is **{jnxLsysSpNATpoipnum 1}**, defines the NAT port overloading IP number-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-natpoipnum.txt .

For information about the enterprise-specific logical system NAT port overloading IP number-specific MIB objects, see the following topics:

- jnxLsysSpNATpoipnumObjects on page 474
- jnxLsysSpNATpoipnumSummary on page 474

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Copyright © 2012, Juniper Networks, Inc.

jnxLsysSpNATpoipnumObjects

The **jnxLsysSpNATpoipnumObjects**, whose object identifier is **{jnxLsysSpNATpoipnumMIB 1}**, represents the logical system NAT port overloading IP number objects as listed in Table 163 on page 474.

Table 163: jnxLsysSpNATpoipnumObjects

Object	Object ID	Description
jnxLsysSpNATpoipnumTable	**jnxLsysSpNATpoipnumObjects 1**	Represents the logical system profile NAT port overloading IP number objects for NAT source pool resource consumption per logical system.
jnxLsysSpNATpoipnumEntry	**jnxLsysSpNATpoipnumTable 1**	Represents an entry in the NAT port overloading IP number resource table.

The **jnxLsysSpNATpoipnumEntry** includes the objects that are listed in Table 164 on page 474.

Table 164: jnxLsysSpNATpoipnumEntry

Object	Object ID	Description
jnxLsysSpNATpoipnumLsysName	**jnxLsysSpNATpoipnumEntry 1**	Represents the name of the logical system for which NAT source pool resource information is retrieved.
jnxLsysSpNATpoipnumProfileName	**jnxLsysSpNATpoipnumEntry 2**	Represents the security profile name string for a logical system.
jnxLsysSpNATpoipnumUsage	**jnxLsysSpNATpoipnumEntry 3**	Represents the current resource usage count for a logical system.
jnxLsysSpNATpoipnumReserved	**jnxLsysSpNATpoipnumEntry 4**	Represents the reserved resource count for a logical system.
jnxLsysSpNATpoipnumMaximum	**jnxLsysSpNATpoipnumEntry 5**	Represents the maximum allowed resource usage count for a logical system.

Related Documentation

- jnxLsysSpNATpoipnumMIB on page 473
- jnxLsysSpNATpoipnumSummary on page 474

jnxLsysSpNATpoipnumSummary

The **jnxLsysSpNATpoipnumSummary**, whose object identifier is **{jnxLsysSpNATpoipnumMIB 2}**, represents the NAT port overloading IP number resource consumption summary.

Table 165 on page 475 lists and describes the NAT source pool resource consumption summary.

Copyright © 2012, Juniper Networks, Inc.

Table 165: jnxLsysSpNATpoipnumSummary

Object	Object ID	Description
jnxLsysSpNATpoipnumUsedAmount	jnxLsysSpNATpoipnumSummary 1	Represents the NAT port overloading IP number resource consumption for all logical systems.
jnxLsysSpNATpoipnumMaxQuota	jnxLsysSpNATpoipnumSummary 2	Represents the NAT port overloading IP number resource maximum quota for all logical systems on the device.
jnxLsysSpNATpoipnumAvailableAmount	jnxLsysSpNATpoipnumSummary 3	Represents the amount of NAT port overloading IP number resources available for all logical systems on the device.
jnxLsysSpNATpoipnumHeaviestUsage	jnxLsysSpNATpoipnumSummary 4	Represents the most used NAT port overloading IP number resources consumed by a logical system.
jnxLsysSpNATpoipnumHeaviestUser	jnxLsysSpNATpoipnumSummary 5	Represents the logical system name that consumes the most NAT port overloading IP number resources.
jnxLsysSpNATpoipnumLightestUsage	jnxLsysSpNATpoipnumSummary 6	Represents the least used NAT port overloading IP number resources consumed by a logical system.
jnxLsysSpNATpoipnumLightestUser	jnxLsysSpNATpoipnumSummary 7	Represents the logical system name that consumes the least NAT port overloading IP number resources.

Related Documentation

- jnxLsysSpNATpoipnumMIB on page 473
- jnxLsysSpNATpoipnumObjects on page 474

LSYS CPU MIB

- jnxLsysSpCPUMIB on page 475
- jnxLsysSpCPUObjects on page 476
- jnxLsysSpCPSummary on page 477
- jnxLsysSpSPUSummary on page 477

jnxLsysSpCPUMIB

The Juniper Networks enterprise-specific LSYS CPU MIB, whose object identifier is **{jnxLsysSpCPU 1}**, defines the CPU-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-lsys-sp-cpu.txt.

For information about the enterprise-specific logical system CPU MIB objects, see the following topics:

- jnxLsysSpCPUObjects on page 476
- jnxLsysSpCPSummary on page 477
- jnxLsysSpSPUSummary on page 477

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxLsysSpCPUObjects

The **jnxLsysSpCPUObjects**, whose object identifier is **{jnxLsysSpCPUMIB 1}**, represents the logical system CPU objects as listed in Table 166 on page 476.

Table 166: jnxLsysSpCPUObjects

Object	Object ID	Description
jnxLsysSpCPUTable	jnxLsysSpCPUObjects 1	Represents the logical system profile objects for CPU resource consumption per logical system.
jnxLsysSpCPUEntry	jnxLsysSpCPUTable 1	Represents an entry in the CPU resource table.

The **JnxLsysSpCPUEntry** includes the objects that are listed in Table 167 on page 476.

Table 167: JnxLsysSpCPUEntry

Object	Object ID	Description
jnxLsysSpCPULsysName	jnxLsysSpCPUEntry 1	Represents the name of the logical system for which CPU resource information is retrieved.
jnxLsysSpCPUProfileName	jnxLsysSpCPUEntry 2	Represents the security profile name string for a logical system.
jnxLsysSpCPUsage	jnxLsysSpCPUEntry 3	Represents the current CPU resource usage count for a logical system.
jnxLsysSpSPUUsage	jnxLsysSpCPUEntry 4	Represents the current SPU resource usage count for a logical system.
jnxLsysSpCPUReserved	jnxLsysSpCPUEntry 5	Represents the reserved resource count for a logical system.
jnxLsysSpCPUMaximum	jnxLsysSpCPUEntry 6	Represents the maximum allowed resource usage count for a logical system.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- jnxLsysSpCPUMIB on page 475
- jnxLsysSpCPSummary on page 477
- jnxLsysSpSPUSummary on page 477

jnxLsysSpCPSummary

The **jnxLsysSpCPSummary**, whose object identifier is **{jnxLsysSpCPUSummary 1}**, represents the CP resource consumption summary.

Table 168 on page 477 lists and describes the CP resource consumption summary.

Table 168: jnxLsysSpCPSummary

Object	Object ID	Description
jnxLsysSpCPUsedAmount	jnxLsysSpCPSummary 1	Represents the CP resource consumption for all logical systems.
jnxLsysSpCPMaxQuota	jnxLsysSpCPSummary 2	Represents the CP resource maximum quota for all logical systems on the device.
jnxLsysSpCPAvailableAmount	jnxLsysSpCPSummary 3	Represents the amount of CP resources available for all logical systems on the device.
jnxLsysSpCPHeaviestUsage	jnxLsysSpCPSummary 4	Represents the most used CP resources consumed by a logical system.
jnxLsysSpCPHeaviestUser	jnxLsysSpCPSummary 5	Represents the logical system name that consumes the most CP resources.
jnxLsysSpCPLightestUsage	jnxLsysSpCPSummary 6	Represents the least used CP resources consumed by a logical system.
jnxLsysSpCPLightestUser	jnxLsysSpCPSummary 7	Represents the logical system name that consumes the least CP resources.

Related Documentation

- jnxLsysSpCPUMIB on page 475
- jnxLsysSpCPUObjects on page 476
- jnxLsysSpSPUSummary on page 477

jnxLsysSpSPUSummary

The **jnxLsysSpSPUSummary**, whose object identifier is **{jnxLsysSpCPUSummary 2}**, represents the SPU resource consumption summary.

Table 169 on page 478 lists and describes the SPU resource consumption summary.

Table 169: jnxLsysSpSPUSummary

Object	Object ID	Description
jnxLsysSpSPUUsedAmount	jnxLsysSpSPUSummary 1	Represents the SPU resource consumption for all logical systems.
jnxLsysSpSPUMaxQuota	jnxLsysSpSPUSummary 2	Represents the SPU resource maximum quota for all logical systems on the device.
jnxLsysSpSPUAvailableAmount	jnxLsysSpSPUSummary 3	Represents the amount of SPU resources available for all logical systems on the device.
jnxLsysSpSPUHeaviestUsage	jnxLsysSpSPUSummary 4	Represents the most used SPU resources consumed by a logical system.
jnxLsysSpSPUHeaviestUser	jnxLsysSpSPUSummary 5	Represents the logical system name that consumes the most SPU resources.
jnxLsysSpSPULightestUsage	jnxLsysSpSPUSummary 6	Represents the least used SPU resources consumed by a logical system.
jnxLsysSpSPULightestUser	jnxLsysSpSPUSummary 7	Represents the logical system name that consumes the least SPU resources.

Related Documentation

- jnxLsysSpCPUMIB on page 475
- jnxLsysSpCPUObjects on page 476
- jnxLsysSpCPSummary on page 477

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 37

Interpreting the Enterprise-Specific Network Address Translation Objects MIB

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484
- NAT Trap Definitions on page 485

NAT Objects MIB

The Juniper Networks enterprise-specific Network Address Translation (NAT) Objects MIB, **jnxJsNatMIB**, whose object ID is **{jnxJsNAT 1}**, defines the objects that are used to monitor NAT attributes.

Related NAT Objects MIBs include:

- **jnxJsNatNotifications {jnxJsNatMIB 0}**
- **jnxJsNatObjects {jnxJsNatMIB 1}**
- **jnxJsNatTrapVars {jnxJsNatMIB 2}**

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-js-nat.txt .

For information about the enterprise-specific NAT Objects MIB, see the following topics:

- Source NAT Table on page 480
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484
- NAT Trap Definitions on page 485

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Source NAT Table

Table 170 on page 480 identifies objects in the Source NAT Table.

Table 170: Source NAT Table

Object	Object ID	Description
jnxJsSrcNatNumOfEntries	**jnxJsNatObjects 1**	Total number of dynamic addresses being translated.
jnxJsSrcNatTable	**jnxJsNatObjects 2**	Table that exposes the source NAT attributes of the translated addresses. When performing source IP address translation, the device translates the original source IP address or port number to a different one. The resource, address source pools, provides the security device with a supply of addresses from which to draw when performing source NAT. The security device has the following types of source pools: • Source pool with Port Address Translation (PAT) • Source pool without PAT • Static source pool This table contains information about source IP address translation only.
jnxJsSrcNatEntry	**jnxJsSrcNatTable 1**	Source NAT address entries. This object is indexed by the address pool table and the address allocated. Sequence of parameters: • **jnxJsNatSrcIpPoolName** • **jnxJsNatSrcGlobalAddr** • **jnxJsNatSrcPortPoolType** • **jnxJsNatSrcNumOfPortInuse** • **jnxJsNatSrcNumOfSessions** • **jnxJsNatSrcAssocatedIf**
jnxJsNatSrcIpPoolName	**jnxJsSrcNatEntry 1**	Address pool from which the translated address is allocated.
jnxJsNatSrcGlobalAddr	**jnxJsSrcNatEntry 2**	Name of dynamic source IP address allocated from the address pool used in the NAT translation.

Copyright © 2012, Juniper Networks, Inc.

Table 170: Source NAT Table *(continued)*

Object	Object ID	Description
jnxJsNatSrcPortPoolType	**jnxJsSrcNatEntry 3**	Source NAT can do address translation with or without PAT. The source port pool type indicates whether the address translation is done with PAT, without PAT, or as a static translation: • **withPAT**—The security device translates both source IP address and port number of the packets. • **withoutPAT**—The device performs source NAT for the IP address without performing PAT for the source port number. • **static**—One range of IP addresses is statically mapped one-to-one to a shifted range of IP addresses.
jnxJsNatSrcNumOfPortInuse	**jnxJsSrcNatEntry 4**	Number of ports in use for this NAT address entry. This attribute is applicable to only NAT translation with PAT.
jnxJsNatSrcNumOfSessions	**jnxJsSrcNatEntry 5**	Number of sessions in use for this NAT address entry. This attribute is applicable to only NAT translation without PAT.
jnxJsNatSrcAssocatedIf	**jnxJsSrcNatEntry 6**	Index of interfaces associated with this NAT address entry. For each interface, the value is a unique value, greater than zero.

Related Documentation

- NAT Objects MIB on page 479
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484
- NAT Trap Definitions on page 485

New Source NAT Table

The **jnxJsSrcNatStatsTable**, whose object ID is **{jnxJsNatObjects 4}**, represents the source NAT attributes of the translated addresses. When performing source IP address translation, the security device translates the original source IP address, the port number, or both to a different address. The resource address source pool provides the security device with a supply of addresses from which to draw when performing source NAT. The new source NAT contains objects on source IP address translation only. Table 171 on page 482 identifies objects in the New Source NAT Table.

Table 171: New Source NAT Table

Object	Object ID	Description
jnxJsSrcNatStatsEntry	jnxJsSrcNatStatsTable 1	Source NAT address entries. This object is indexed by the address pool table and the allocated address. Sequence of parameters: • jnxJsNatSrcPoolName • jnxJsNatSrcXlatedAddrType • jnxJsNatSrcXlatedAddr • jnxJsNatSrcPoolType • jnxJsNatSrcNumPortInuse • jnxJsNatSrcNumSessions
jnxJsNatSrcPoolName	jnxJsSrcNatStatsEntry 1	The name of the dynamic source IP address pool. This is the address pool from which the translated address is allocated.
jnxJsNatSrcXlatedAddrType	jnxJsSrcNatStatsEntry 2	The type of the dynamic source IP address allocated from the address pool used in the NAT translation. The MIB supports IPv4 and IPv6 addresses only.
jnxJsNatSrcXlatedAddr	jnxJsSrcNatStatsEntry 3	The name of the dynamic source IP address allocated from the address pool used in the NAT translation. The MIB supports IPv4 and IPv6 addresses only.
jnxJsNatSrcPoolType	jnxJsSrcNatStatsEntry 4	Source NAT can do address translation with or without Port Address Translation (PAT). The following source port pool type indicates whether the address translation is done with PAT, without PAT, or as a static translation: • **Source pool with PAT**—The security device translates both the source IP address and the port number of the packets. • **Source pool without PAT**—The device performs source NAT translation for the IP address without performing PAT for the source port number. • **Static**—One range of IP addresses is statically mapped one-to-one to a shifted range of IP addresses.
jnxJsNatSrcNumPortInuse	jnxJsSrcNatStatsEntry 5	The number of ports in use for this NAT address entry. This attribute is applicable only to NAT translation with PAT.
jnxJsNatSrcNumSessions	jnxJsSrcNatStatsEntry 6	The number of sessions in use based on this NAT address entry. This attribute is applicable only to NAT translation without PAT.

Related Documentation

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484

Copyright © 2012, Juniper Networks, Inc.

- NAT Trap Definitions on page 485

jnxJsNatIfSrcPoolPortTable

jnxJsNatIfSrcPoolPortTable, whose object ID is **{jnxJsNatObjects 3}**, monitors the port usage of the NAT interface source IP address pool by displaying information about the objects listed in Table 172 on page 483.

The interface source pool is predefined. This source pool is referenced in a policy in which it is configured. The security device translates the source IP address to the address of the egress interface for the traffic, matching a policy that references the interface source pool. The security device always applies PAT for the interface source pool.

Table 172: jnxJsNatIfSrcPoolPortTable

Object	Object ID	Description
jnxJsNatIfSrcPoolPortEntry	**jnxJsNatIfSrcPoolPortTable 1**	Source NAT address entries. This object is indexed by the address pool table and the address. Sequence of parameters: • **jnxJsNatIfSrcPoolIndex** • **jnxJsNatIfSrcPoolTotalSinglePorts** • **jnxJsNatIfSrcPoolAllocSinglePorts** • **jnxJsNatIfSrcPoolTotalTwinPorts** • **jnxJsNatIfSrcPoolAllocTwinPorts**
jnxJsNatIfSrcPoolIndex	**jnxJsNatIfSrcPoolPortEntry 1**	Index number of the port pool of this address pool.
jnxJsNatIfSrcPoolTotalSinglePorts	**jnxJsNatIfSrcPoolPortEntry 2**	Total number of single ports in a port pool.
jnxJsNatIfSrcPoolAllocSinglePorts	**jnxJsNatIfSrcPoolPortEntry 3**	Number of single ports in a port pool allocated or in use.
jnxJsNatIfSrcPoolTotalTwinPorts	**jnxJsNatIfSrcPoolPortEntry 4**	Total number of twin ports in a port pool.
jnxJsNatIfSrcPoolAllocTwinPorts	**jnxJsNatIfSrcPoolPortEntry 5**	Number of twin ports in a port pool allocated or in use.

Related Documentation

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- New Source NAT Table on page 481
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484
- NAT Trap Definitions on page 485

Copyright © 2012, Juniper Networks, Inc.

NAT Rule Hit Table

The **jnxJsNatRuleTable**, whose object ID is **{jnxJsNatObjects 5}**, monitors the NAT rule hits. Table 173 on page 484 identifies objects in the NAT Rule Hit Table.

Table 173: NAT Rule Hit Table

Object	Object ID	Description
jnxJsNatRuleEntry	**jnxJsNatRuleTable 1**	NAT rule hit entries. This object is indexed by the rule index. Sequence of parameters: • **jnxJsNatRuleName** • **jnxJsNatRuleType** • **jnxJsNatRuleTransHits**
jnxJsNatRuleName	**jnxJsNatRuleEntry 1**	NAT rule name.
jnxJsNatRuleType	**jnxJsNatRuleEntry 2**	NAT types such as source, destination, and static.
jnxJsNatRuleTransHits	**jnxJsNatRuleEntry 3**	Number of hits on this NAT rule.

Related Documentation

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Pool Hit Table on page 484
- NAT Trap Definitions on page 485

NAT Pool Hit Table

The **jnxJsNatPoolTable**, whose object ID is **{jnxJsNatObjects 6}**, monitors the NAT pool hits. Table 174 on page 484 identifies objects in the NAT Pool Hit Table.

Table 174: NAT Pool Hit Table

Object	Object ID	Description
jnxJsNatPoolEntry	**jnxJsNatPoolTable 1**	NAT pool hit entries. This object is indexed by the pool index. Sequence of parameters: • **jnxJsNatPoolName** • **jnxJsNatPoolType** • **jnxJsNatPoolTransHits**
jnxJsNatPoolName	**jnxJsNatPoolEntry 1**	NAT pool name.

Copyright © 2012, Juniper Networks, Inc.

Table 174: NAT Pool Hit Table *(continued)*

Object	Object ID	Description
jnxJsNatPoolType	**jnxJsNatPoolEntry 2**	NAT types such as source, destination, and static.
jnxJsNatPoolTransHits	**jnxJsNatPoolEntry 3**	Number of hits on this NAT pool.

Related Documentation

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Trap Definitions on page 485

NAT Trap Definitions

Table 175 on page 485 lists NAT trap definition objects.

Table 175: NAT Trap Definitions

Object	Object ID	Description
jnxJsNatAddrPoolThresholdStatus	**jnxJsNatNotifications 1**	NAT address pool utilization threshold status trap to signify that the address pool utilization either exceeds a certain percentage or is clear of that percentage. **jnxJsNatSrcIpPoolName** is the name of the resource pool. **jnxJsNatAddrPoolUtil** is the percentage of utilization of the address pool.
jnxJsNatAddrPoolUtil	**jnxJsNatTrapVars 1**	Dynamic address pool utilization expressed as a percentage.

Related Documentation

- NAT Objects MIB on page 479
- Source NAT Table on page 480
- New Source NAT Table on page 481
- jnxJsNatIfSrcPoolPortTable on page 483
- NAT Rule Hit Table on page 484
- NAT Pool Hit Table on page 484

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 38

Interpreting the Enterprise-Specific Packet Forwarding Engine MIB

- Packet Forwarding Engine MIB on page 487
- jnxPfeNotifyGlTable on page 487
- jnxPfeNotifyTypeTable on page 489

Packet Forwarding Engine MIB

The Juniper Networks enterprise-specific Packet Forwarding Engine MIB, whose object identifier is **{jnxPfeMibRoot 1}**, provides notification statistics for Packet Forwarding Engines. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-pfe.txt .

NOTE: **Information provided by this MIB is modeled after information provided by the show pfe statistics notification command-line interface (CLI) command.**

For information about the enterprise-specific Packet Forwarding Engine MIB objects, see the following topics:

- jnxPfeNotifyGlTable on page 487
- jnxPfeNotifyTypeTable on page 489

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxPfeNotifyGlTable

jnxPfeNotifyGlTable contains global Packet Forwarding Engine notification statistics.

JnxPfeNotifyGlEntry contains notification statistics for each Packet Forwarding Engine slot. Each entry has objects listed in Table 176 on page 488.

Table 176: jnxPfeNotifyGlEntry

Object	Object Identifier	Description
jnxPfeNotifyGISlot	jnxPfeNotifyGIEntry 1	The slot number for a set of Packet Forwarding Engine notification statistics.
jnxPfeNotifyGIParsed	jnxPfeNotifyGIEntry 2	The number of notifications reported by the Packet Forwarding Engine controller, which manages packet forwarding functions.
jnxPfeNotifyGIAged	jnxPfeNotifyGIEntry 3	The number of notifications dropped because they have expired.
jnxPfeNotifyGICorrupt	jnxPfeNotifyGIEntry 4	The number of notifications dropped because the notification result format is invalid. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGIIllegal	jnxPfeNotifyGIEntry 5	The number of notifications dropped because the notification has an illegal notification type.
jnxPfeNotifyGISample	jnxPfeNotifyGIEntry 6	The number of sample notifications reported by the Packet Forwarding Engine controller.
jnxPfeNotifyGIGiants	jnxPfeNotifyGIEntry 7	The number of notifications dropped because the notification is larger than the supported direct memory access (DMA) size.
jnxPfeNotifyGITtlExceeded	jnxPfeNotifyGIEntry 8	The number of options/TTL-expired notifications sent to service interfaces as transit packets. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGITtlExcErrors	jnxPfeNotifyGIEntry 9	The number of options/TTL-expired notifications that could not be sent to service interfaces as transit packets because the output interface could not be determined. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGISvcOptAsp	jnxPfeNotifyGIEntry 10	The number of IP options packets sent to a Services PIC.
jnxPfeNotifyGISvcOptRe	jnxPfeNotifyGIEntry 11	The number of IP options packets sent to the Routing Engine.
jnxPfeNotifyGIPostSvcOptOut	jnxPfeNotifyGIEntry 12	The number of notifications re-injected by a Services PIC after processing the associated packets. The Packet Forwarding Engine will forward these notifications to their actual destination. This counter is valid for Internet Processor-I and Internet Processor-II only.
jnxPfeNotifyGIOptTtlExp	jnxPfeNotifyGIEntry 13	The number of TTL-expired transit packets.
jnxPfeNotifyGIDiscSample	jnxPfeNotifyGIEntry 14	The number of sample notifications dropped because the notifications refer to discarded packets in the Packet Forwarding Engine.
jnxPfeNotifyGIRateLimited	jnxPfeNotifyGIEntry 15	The number of notification ignored because of Packet Forwarding Engine software throttling (delaying or refusing requests).
jnxPfeNotifyGIPktGetFails	jnxPfeNotifyGIEntry 16	The number of notifications ignored because DMA memory could not be allocated.

Copyright © 2012, Juniper Networks, Inc.

Table 176: jnxPfeNotifyGlEntry *(continued)*

Object	Object Identifier	Description
jnxPfeNotifyGlDmaFails	**jnxPfeNotifyGlEntry 17**	The number of notifications where the DMA of associated packets failed for miscellaneous reasons. This counter is valid for T Series routers only.
jnxPfeNotifyGlDmaTotals	**jnxPfeNotifyGlEntry 18**	The number of notifications for which the packet DMA completed. This counter is valid for T Series routers only.
jnxPfeNotifyGlUnknowns	**jnxPfeNotifyGlEntry 19**	The number of notifications that could not be resolved to a known, next hop destination. This counter is valid for T Series routers only.

Related Documentation

- Packet Forwarding Engine MIB on page 487
- jnxPfeNotifyTypeTable on page 489

jnxPfeNotifyTypeTable

jnxPfeNotifyTypeTable contains information about type-specific Packet Forwarding Engine notifications for each Packet Forwarding Engine slot. Packet Forwarding Engine notification types are listed in Table 177 on page 489.

Table 177: Packet Forwarding Engine Notification Types

Index	Item	Description
1	Illegal	Packets with an invalid notification type.
2	Unclassified	Packets that did not have a key lookup performed on them.
3	Option	Packets that include L3 options.
4	Next Hop	Packets that are destined to the host.
5	Discard	Discarded packets sent to the route processor.
6	Sample	Unused.
7	Redirect	Packets sent back to the interfaces from which they arrived.
8	Do Not Fragment	Packets that need to be fragmented, but have a **don't fragment (DF)** value set.
9	CFDF	Packets that have a DF value set and have triggered a maximum transmission unit (MTU) exceeded indicator.
10	Poison	Packets that have a poisoned next-hop index.

For information, see **jnxPfeNotifyTypeEntry**.

jnxPfeNotifyTypeEntry

jnxPfeNotifyTypeEntry contains information about type-specific Packet Forwarding Engine notifications, and has the objects listed in Table 178 on page 490.

Table 178: jnxPfeNotifyTypeEntry

Object	Object Identifier	Description
jnxPfeNotifyTypeId	**jnxPfeNotifyTypeEntry 1**	Identifies the Packet Forwarding Engine notification type. See Table 177 on page 489 for a list of notification types.
jnxPfeNotifyTypeDescr	**jnxPfeNotifyTypeEntry 2**	A description of the Packet Forwarding Engine notification type.
jnxPfeNotifyTypeParsed	**jnxPfeNotifyTypeEntry 3**	The number of notifications that are parsed successfully.
jnxPfeNotifyTypeInput	**jnxPfeNotifyTypeEntry 4**	The number of notifications whose associated packets are stored in router processor memory using direct memory access.
jnxPfeNotifyTypeFailed	**jnxPfeNotifyTypeEntry 5**	The number of notifications that are not parsed successfully.
jnxPfeNotifyTypeIgnored	**jnxPfeNotifyTypeEntry 6**	The number of notifications where the notification type in the message does not match any of the valid notification types.

Related Documentation

- Packet Forwarding Engine MIB on page 487
- jnxPfeNotifyGlTable on page 487

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 39

Interpreting the Enterprise-Specific Ping MIB

- PING MIB on page 491
- jnxPingCtlTable on page 492
- jnxPingResultsTable on page 496
- jnxPingProbeHistoryTable on page 500
- jnxPingLastTestResultTable on page 502

PING MIB

The Juniper Networks enterprise-specific Ping MIB extends the standard Ping MIB control table (RFC 2925). The Ping MIB, whose object identifier is **{jnxMIbs 7}**, allows you to monitor network delay (latency), packet loss, network delay variation (jitter), one-way latency, and other network statistics.

Items in this MIB are created when entries are created in the **pingCtlTable** of the Ping MIB. Each item is indexed exactly as in the Ping MIB.

To view a complete copy of the enterprise-specific extensions to the Ping MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-ping.txt .

For information about using the Ping MIB and enterprise-specific Ping MIB, see the *Junos OS Network Management Configuration Guide*.

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

- jnxPingCtlTable on page 492
- jnxPingResultsTable on page 496
- jnxPingProbeHistoryTable on page 500
- jnxPingLastTestResultTable on page 502

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxPingCtlTable

The enterprise-specific Ping MIB structure includes one main object, **jnxPingCtlTable**, whose object identifier is **{jnxPingObjects 2}**, and defines the **jnxPing** control table for providing enterprise-specific options to the corresponding **pingCtlEntry**. **jnxPingCtlTable** monitors thresholds; for example, the maximum allowed jitter in the trip time during a text.

Each **jnxPingCtlEntry** has two indexes identical to those of the corresponding **pingCtlEntry**. Entries created in **pingCtlTable** are mirrored here. **jnxPingCtlTable** objects are listed in Table 179 on page 492.

Table 179: jnxPingCtlTable

Object	Object Identifier	Description
jnxCtlOwnerIndex	**jnxPingCtlEntry 1**	The first index. It is identical to the **pingCtlOwnerIndex** of the corresponding **pingCtlEntry** in the **pingCtlTable**.
jnxPingCtlTestName	**jnxPingCtlEntry 2**	The other index and is identical to the **pingCtlTestName** of the corresponding **pingCtlEntry** in the **pingCtlTable**.
jnxPingCtlIfName	**jnxPingCtlEntry 3**	Specifies the name of the outgoing interface for ping probes. This is the name-based complement to **pingCtlIfIndex**. A zero-length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: • **pingCtlIfIndex** (see **pingCtlTable** in "PING MIB" on page 491) • **jnxPingCtlIfName** • **jnxPingCtlRoutingInstanceName**
jnxPingCtlRoutingInstanceName	**jnxPingCtlEntry 6**	Specifies the name of the routing instance used when directing outgoing ping packets. The instance name specified must be configured at the **[edit routing-instances]** hierarchy level of the Junos configuration. The instance-type must be **vrf**.
jnxPingCtlRttThreshold	**jnxPingCtlEntry 7**	The maximum round-trip time allowed. If this threshold is crossed by any probe, a **jnxPingRttThresholdExceeded** trap is sent.

Copyright © 2012, Juniper Networks, Inc.

Table 179: jnxPingCtlTable *(continued)*

Object	Object Identifier	Description
jnxPingCtlRttStdDevThreshold	**jnxPingCtlEntry 8**	The maximum round-trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the round-trip time at the end of any test exceeds this threshold, a **jnxPingRttStdDevThresholdExceeded** trap is sent.
jnxPingCtlRttJitterThreshold	**jnxPingCtlEntry 9**	The maximum allowed jitter in the round-trip time over the course of any test. Jitter is the difference between the maximum and minimum round-trip times measured over the course of a single test (**jnxPingResultsMaxRttUs** minus **jnxPingResultsMinRttUs**). If the measured jitter exceeds this threshold, a **jnxPingRttJitterThresholdExceeded** trap is sent.
jnxPingCtlEgressTimeThreshold	**jnxPingCtlEntry 10**	Maximum egress trip time allowed. If this threshold is crossed by any probe, a **jnxPingEgressThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. Currently, **jnxPingIcmpTimeStamp** is the only supported probe type with this property.
jnxPingCtlEgressStdDevThreshold	**jnxPingCtlEntry 11**	The maximum egress trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the egress trip time at the end of any test exceeds this threshold, a **jnxPingEgressStdDevThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one way delay measurements. The **jnxPingIcmpTimeStamp** is the only supported probe type with this property.
jnxPingCtlEgressJitterThreshold	**jnxPingCtlEntry 12**	The maximum allowed jitter in the egress trip time over the course of any test. Jitter is defined as the difference between the maximum and minimum egress trip times measured over the course of a single test (**jnxPingResultsMaxSrcDstt** minus **jnxPingResultsMinSrcDstt**). If the measured jitter exceeds this threshold, a **jnxPingEgressJitterThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. The **jnxPingIcmpTimeStamp** is the only supported probe type with this property.

Copyright © 2012, Juniper Networks, Inc.

Table 179: jnxPingCtlTable *(continued)*

Object	Object Identifier	Description
jnxPingCtlIngressTimeThreshold	**jnxPingCtlEntry 13**	The maximum ingress trip time allowed. If this threshold is crossed by any probe, a **jnxPingIngressThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. The **jnxPingIcmpTimeStamp** is the only supported probe type with this property.
jnxPingCtlIngressStddevThreshold	**jnxPingCtlEntry 14**	The maximum ingress trip time standard deviation allowed over the course of any test. If the calculated standard deviation of the ingress trip time at the end of any test exceeds this threshold, a **jnxPingIngressStddevThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. Currently, **jnxPingIcmpTimeStamp** is the only supported probe type with this property.
jnxPingCtlIngressJitterThreshold	**jnxPingCtlEntry 15**	The maximum allowed jitter in the ingress trip time over the course of any test. Jitter is defined as the difference between the maximum and minimum ingress trip times measured over the course of a single test (**jnxPingResultsMaxDstSrct** minus **jnxPingResultsMinDstSrct**). If the measured jitter exceeds this threshold, a **jnxPingIngressJitterThresholdExceeded** trap is sent. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. The **jnxPingIcmpTimeStamp** is the only supported probe type with this property.

Copyright © 2012, Juniper Networks, Inc.

Table 179: jnxPingCtlTable *(continued)*

Object	Object Identifier	Description
jnxPingCtlTrapGeneration	**jnxPingCtlEntry 16**	The value of this object determines when and if to generate a notification for this entry. **rttThreshold(0)**—Generate a **jnxPingRttThresholdExceeded** notification when the configured RTT threshold is exceeded. **rttStdDevThreshold(1)**—Generate a **jnxPingRttStdDevThresholdExceeded** notification when the configured RTT standard deviation threshold is exceeded. **rttJitterThreshold(2)**—Generate a **jnxPingRttJitterThresholdExceeded** notification when the configured RTT jitter threshold is exceeded. **egressThreshold(3)**—Generate a **jnxPingEgressThresholdExceeded** notification when the configured egress threshold is exceeded. This applies only if the probe type supports one-way measurements.

Copyright © 2012, Juniper Networks, Inc.

Table 179: jnxPingCtlTable *(continued)*

Object	Object Identifier	Description
		egressStdDevThreshold(4)—Generate a **jnxPingEgressStdDevThresholdExceeded** notification when the configured egress standard deviation threshold is exceeded. This applies only if the probe type supports one-way measurements. **egressJitterThreshold(5)**—Generate a **jnxPingEgressJitterThresholdExceeded** notification when the configured egress jitter threshold is exceeded. This applies only if the probe type supports one-way measurements. **ingressThreshold(6)**—Generate a **jnxPingIngressThresholdExceeded** notification when the configured ingress threshold is exceeded. This applies only if the probe type supports one-way measurements. **ingressStdDevThreshold(7)**—Generate a **jnxPingIngressStdDevThresholdExceeded** notification when the configured ingress standard deviation threshold is exceeded. This applies only if the probe type supports one way measurements. **ingressJitterThreshold(8)**—Generate a **jnxPingIngressJitterThresholdExceeded** notification when the configured ingress jitter threshold is exceeded. This applies only if the probe type supports one-way measurements. The value of this object defaults to zero, indicating that none of the above options have been selected.

Related Documentation

- PING MIB on page 491
- jnxPingResultsTable on page 496
- jnxPingProbeHistoryTable on page 500
- jnxPingLastTestResultTable on page 502

jnxPingResultsTable

jnxPingResultsTable, whose object identifier is **{jnxPingObjects 3}**, gathers ping test results on traffic on round-trip, ingress, and egress trip delays. This is useful when you want to measure the performance of your network and verify service-level agreements with your vendors.

The **jnxPingResultsEntry** objects are listed in Table 180 on page 497.

Copyright © 2012, Juniper Networks, Inc.

Table 180: jnxPingsResultsEntry

Object	Object Identifier	Description
jnxPingResultsRttUs	**jnxPingResultsEntry 1**	The round-trip delays measured for the most recent successful probe during this test, in microseconds.
jnxPingResultsSumRttUs	**jnxPingResultsEntry 2**	The sum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsMinRttUs	**jnxPingResultsEntry 3**	The minimum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsMaxRttUs	**jnxPingResultsEntry 4**	The maximum of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsAvgRttUs	**jnxPingResultsEntry 5**	The average of the round-trip delays measured for all the probes during this test, in microseconds.
jnxPingResultsStdDevRttUs	**jnxPingResultsEntry 6**	The standard deviation of the round-trip delays measured during this test, in microseconds.
jnxPingResultsEgressUs	**jnxPingResultsEntry 7**	The egress trip delays measured for the most recent successful probe during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsMinEgressUs	**jnxPingResultsEntry 8**	The minimum of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsMaxEgressUs	**jnxPingResultsEntry 9**	The maximum of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.

Copyright © 2012, Juniper Networks, Inc.

Table 180: jnxPingsResultsEntry *(continued)*

Object	Object Identifier	Description
jnxPingResultsAvgEgressUs	**jnxPingResultsEntry 10**	The average of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsStddevEgressUs	**jnxPingResultsEntry 11**	The standard deviation of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsIngressUs	**jnxPingResultsEntry 12**	The ingress trip delays measured for the most recent successful probe during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsMinIngressUs	**jnxPingResultsEntry 13**	The minimum of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsMaxIngressUs	**jnxPingResultsEntry 14**	The maximum of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsAvgIngressUs	**jnxPingResultsEntry 15**	The average of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.

Copyright © 2012, Juniper Networks, Inc.

Table 180: jnxPingsResultsEntry *(continued)*

Object	Object Identifier	Description
jnxPingResultsStddevIngressUs	jnxPingResultsEntry 16	The standard deviation of the ingress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsJitterRttUs	jnxPingResultsEntry 17	The jitter of the round-trip delays measured for all probes during this test, in microseconds.
jnxPingResultsJitterEngressUs	jnxPingResultsEntry 18	The jitter of the engress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsJitterIngressUs	jnxPingResultsEntry 19	The jitter of the ingress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingResultsStatus	jnxPingResultsEntry 20	The result of the most recent probe.
jnxPingResultsTime	jnxPingResultsEntry 21	The date and time of the most recent probe result.
jnxPingResultsOwnerIndex	jnxPingResultsEntry 22	The first index. It has the same value as **pingCtlOwnerIndex** and is provided for applications that are unable to obtain the value of **pingCtlOwnerIndex** from the instance portion of the object identifiers belonging to this table.
jnxPingResultsTestName	jnxPingResultsEntry 23	The other index. It has the same value as **pingCtlTestName** and is provided for applications that are unable to obtain the value of **pingCtlTestName** from the instance portion of the object identifiers belonging to this table.

Related Documentation

- PING MIB on page 491
- jnxPingCtlTable on page 492
- jnxPingProbeHistoryTable on page 500

- jnxPingLastTestResultTable on page 502

jnxPingProbeHistoryTable

jnxpingProbeHistoryTable, whose object identifier is **{jnxPingObjects 4}**, contains the history of all ping tests.

The **jnxPingProbeHistoryEntry** objects are listed in Table 181 on page 500.

Table 181: jnxPingProbeHistoryEntry

Object	Object Identifier	Description
jnxPingProbeHistoryResponseUs	**jnxPingProbeHistoryEntry 1**	The amount of time, in microseconds, from when a probe was sent to when its response was received or when it timed out. The value of this object is reported as 0 when it is not possible to transmit a probe.
jnxPingProbeHistoryJitterUs	**jnxPingProbeHistoryEntry 2**	The time difference, in microseconds, between the maximum and minimum round-trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed.
jnxPingProbeHistoryResponseEgressUs	**jnxPingProbeHistoryEntry 3**	The amount of time, in microseconds, from when a probe was sent to when it was received by destination. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, the value is irrelevant and this object returns 0.

Copyright © 2012, Juniper Networks, Inc.

Table 181: jnxPingProbeHistoryEntry *(continued)*

Object	Object Identifier	Description
jnxPingProbeHistoryResponseIngressUs	jnxPingProbeHistoryEntry 4	The amount of time, in microseconds, from when a probe was sent from the destination to when it was received. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, the value is irrelevant and this object returns 0.
jnxPingProbeHistoryEgressJitterUs	jnxPingProbeHistoryEntry 5	The time difference, in microseconds, between the maximum and minimum egress trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, the value is irrelevant and this object returns 0.
jnxPingProbeHistoryIngressJitterUs	jnxPingProbeHistoryEntry 6	The time difference, in microseconds, between the maximum and minimum ingress trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, the value is irrelevant and this object returns 0.

Related Documentation

- PING MIB on page 491
- jnxPingCtlTable on page 492
- jnxPingResultsTable on page 496

• jnxPingLastTestResultTable on page 502

jnxPingLastTestResultTable

jnxPingLastTestResultTable, whose object identifier is **{jnxPingObjects 5}**, contains the results of the last completed ping tests. Entries corresponding to a test are created only after completion of the first test. This is useful when you want to ensure that a test has been completed before collecting test results.

The **jnxPingLastTestResultEntry** objects are listed in Table 182 on page 502.

Table 182: jnxPingLastTestResultEntry

Object	Object Identifier	Description
jnxPingLastTestResultProbeResponses	jnxPingLastTestResultEntry 1	The number of responses received in the most recently completed test.
jnxPingLastTestResultSentProbes	jnxPingLastTestResultEntry 2	The number of probes sent in the most recently completed test.
jnxPingLastTestResultSumRttUs	jnxPingLastTestResultEntry 3	The sum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinRttUs	jnxPingLastTestResultEntry 4	The minimum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMaxRttUs	jnxPingLastTestResultEntry 5	The maximum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultAvgRttUs	jnxPingLastTestResultEntry 6	The average of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.

Copyright © 2012, Juniper Networks, Inc.

Table 182: jnxPingLastTestResultEntry *(continued)*

Object	Object Identifier	Description
jnxPingLastTestResultStdDevRttUs	**jnxPingLastTestResultEntry 7**	The standard deviation of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinEgressUs	**jnxPingLastTestResultEntry 8**	The minimum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultMaxEgressUs	**jnxPingLastTestResultEntry 9**	The maximum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultAvgEgressUs	**jnxPingLastTestResultEntry 10**	The average of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.

Copyright © 2012, Juniper Networks, Inc.

Table 182: jnxPingLastTestResultEntry *(continued)*

Object	Object Identifier	Description
jnxPingLastTestResultStddevEgressUs	**jnxPingLastTestResultEntry 11**	The standard deviation of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultMinIngressUs	**jnxPingLastTestResultEntry 12**	The minimum of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultMaxIngressUs	**jnxPingLastTestResultEntry 13**	The maximum of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultAvgIngressUs	**jnxPingLastTestResultEntry 14**	The average of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.

Copyright © 2012, Juniper Networks, Inc.

Table 182: jnxPingLastTestResultEntry *(continued)*

Object	Object Identifier	Description
jnxPingLastTestResultStddevIngressUs	jnxPingLastTestResultEntry 15	The standard deviation of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResult PeakToPeakJitterRttUs	jnxPingLastTestResultEntry 16	The difference between the minimum and maximum delays over the course of the last completed test, in microseconds.
jnxPingLastTestResult PeakToPeakJitterEgressUs	jnxPingLastTestResultEntry 17	The difference between the minimum and maximum egress trip delays over the course of the last completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResult PeakToPeakJitterIngressUs	jnxPingLastTestResultEntry 18	The difference between the minimum and maximum ingress trip delays over the course of the last completed test, in microseconds. This applies only if the probe type (**pingCtlType**) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.
jnxPingLastTestResultTime	jnxPingLastTestResultEntry 19	The time the last test was completed.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- PING MIB on page 491
- jnxPingCtlTable on page 492
- jnxPingResultsTable on page 496
- jnxPingProbeHistoryTable on page 500

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 40

Interpreting the Enterprise-Specific Policy Objects MIB

- Policy Objects MIB on page 507
- Security Policy Table on page 508
- jnxJsPolicyStatsTable on page 509

Policy Objects MIB

The Juniper Networks enterprise-specific Policy Objects MIB, **jnxJsSecPolicyMIB**, whose object ID is **{jnxJsPolicies 1}**, defines the MIB for policy monitoring.

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining one or more kinds of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

The Juniper Networks security device enforces the security policies rules for the transit traffic in terms of which traffic can pass through the firewall and the actions taken on the traffic as it passes through the firewall.

Related MIB objects include the following:

- **jnxJsPolicyNotifications—{jnxJsSecPolicyMIB 0}**
- **jnxJsPolicyObjects—{jnxJsSecPolicyMIB 1}**
- **jnxJsPolicyTrapVars—{jnxJsSecPolicyMIB 2}**

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-policy.txt .

For information about the enterprise-specific Policy Objects MIB, see the following topics:

- Security Policy Table on page 508
- jnxJsPolicyStatsTable on page 509

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

Security Policy Table

Table 183 on page 508 describes the objects in the Security Policy Table.

Table 183: Security Policy Table

Object	Object ID	Description
jnxJsPolicyNumber	**jnxJsPolicyObjects 1**	Number of policies (regardless of their current state) present on this system.
jnxJsPolicyTable	**jnxJsPolicyObjects 2**	Exposes the security policy entries. Security devices and routers provide a network boundary with a single point of entry and exit, which allows the screening and directing of traffic through the implementation of access policies. The access policies can permit, deny, encrypt, authenticate, prioirtize, schedule, and monitor the traffic flow through the firewall. This table lists entries of policy. The number of policies are indicated by **jnxJsPolicyNumber**.
jnxJsPolicyEntry	**jnxJsPolicyTable 1**	An entry contains a security policy. Indexes: • **nxJsPolicyFromZone** • **jnxJsPolicyToZone** • **jnxJsPolicyName** Security policies are configured under FromZone and ToZone directions. Under a specific zone direction, each security policy contains a name, match-criteria, action, and other options. Sequence of parameters: • **jnxJsPolicyFromZone** • **jnxJsPolicyToZone** • **jnxJsPolicyName** • **jnxJsPolicySequenceNumber** • **jnxJsPolicyAction** • **jnxJsPolicyScheduler** • **jnxJsPolicyState** • **jnxJsPolicyStatsAvailability** • **jnxJsPolicyPerSecBytesThreshold** • **jnxJsPolicyPerMinKbytesThreshold**
jnxJsPolicyFromZone	**jnxJsPolicyEntry 1**	FromZone name.

Copyright © 2012, Juniper Networks, Inc.

Table 183: Security Policy Table *(continued)*

Object	Object ID	Description
jnxJsPolicyToZone	jnxJsPolicyEntry 2	ToZone name.
jnxJsPolicyName	jnxJsPolicyEntry 3	Name of the policy defined. The name consists of up to 256 ASCII characters and uniquely identifies the policy entry.
jnxJsPolicySequenceNumber	jnxJsPolicyEntry 4	Indication of the policy sequence order of the policy within a specific FromZone and ToZone pair. Policies are matched in a sequence in which the ordering is specified by this number.
jnxJsPolicyAction	jnxJsPolicyEntry 5	Indication of the actions performed when the criteria are matched. The actions **permit**, **reject**, and **deny** are user-configured policies.
jnxJsPolicyScheduler	jnxJsPolicyEntry 6	Name of the schedule attached to this policy. Certain schedules have a specified duration that can affect the status of the policy.
jnxJsPolicyState	jnxJsPolicyEntry 7	State of this policy: active, inactive, or unavailable. The state can be affected by the scheduler if the scheduler has a specified duration.
jnxJsPolicyStatsAvailability	jnxJsPolicyEntry 8	Indication of whether the statistics counters are available and are actively updated. If available, a matching **jnxJsPolicyStatsEntry** exists for the policy.
jnxJsPolicyPerSecBytesThreshold	jnxJsPolicyEntry 9	Indication of the threshold value of bytes per second
jnxJsPolicyPerMinKbytesThreshold	jnxJsPolicyEntry 10	Indication of the threshold value of KB per minute

Related Documentation

- Policy Objects MIB on page 507
- jnxJsPolicyStatsTable on page 509

jnxJsPolicyStatsTable

jnxJsPolicyStatsTable, whose object ID is **{jnxJsPolicyObjects 3}**, exposes the security policy statistics entries listed in Table 184 on page 510. These statistics can be enabled and disabled by configuration on a per-policy basis.

Table 184: jnxJsPolicyStatsTable

Object	Object ID	Description
jnxJsPolicyStatsEntry	**jnxJsPolicyStatsTable 1**	Contains security policy statistics. Indexes: • **jnxJsPolicyFromZone** • **jnxJsPolicyToZone** • **jnxJsPolicyName** Security policies are configured under FromZone and ToZone direction. Under a specific zone direction, each security policy contains name, match-criteria, action, and other options. Sequence of parameters: • **jnxJsPolicyStatsCreationTime** • **jnxJsPolicyStatsInputBytes** • **jnxJsPolicyStatsInputByteRate** • **jnxJsPolicyStatsOutputBytes** • **jnxJsPolicyStatsOutputByteRate** • **jnxJsPolicyStatsInputPackets** • **jnxJsPolicyStatsInputPacketRate** • **jnxJsPolicyStatsOutputPackets** • **jnxJsPolicyStatsOutputPacketRate** • **jnxJsPolicyStatsNumSessions** • **jnxJsPolicyStatsSessionRate** • **jnxJsPolicyStatsSessionDeleted** • **jnxJsPolicyStatsLookups** • **jnxJsPolicyStatsCountAlarm**
jnxJsPolicyStatsCreationTime	**jnxJsPolicyStatsEntry 1**	Creation timestamp of the policy statistics entry. The timestamp is modified during the creation and deletion of the policy statistics entry. When the timestamp changes, the policy entry statistics entry is assumed to be a new statistics entry and not associated with a previous statistic entry of the same indices.
jnxJsPolicyStatsInputBytes	**jnxJsPolicyStatsEntry 2**	Number of input bytes that enter the firewall through this policy.
jnxJsPolicyStatsInputByteRate	**jnxJsPolicyStatsEntry 3**	Number of input bytes per second or the rate that enters the firewall through this policy.
jnxJsPolicyStatsOutputBytes	**jnxJsPolicyStatsEntry 4**	Number of output bytes associated with this policy.

Copyright © 2012, Juniper Networks, Inc.

Table 184: jnxJsPolicyStatsTable *(continued)*

Object	Object ID	Description
jnxJsPolicyStatsOutputByteRate	jnxJsPolicyStatsEntry 5	Number of output bytes per second or the rate associated with this policy.
jnxJsPolicyStatsInputPackets	jnxJsPolicyStatsEntry 6	Number of input packets that enter the firewall through this policy.
jnxJsPolicyStatsInputPacketRate	jnxJsPolicyStatsEntry 7	Number of input packets per second or the input packet rate of the firewall through this policy.
jnxJsPolicyStatsOutputPackets	jnxJsPolicyStatsEntry 8	Number of output packets associated with this policy.
jnxJsPolicyStatsOutputPacketRate	jnxJsPolicyStatsEntry 9	Number of output packets per second or the rate associated with this policy.
jnxJsPolicyStatsNumSessions	jnxJsPolicyStatsEntry 10	Number of sessions associated with this policy.
jnxJsPolicyStatsSessionRate	jnxJsPolicyStatsEntry 11	Rate of the sessions associated with this policy.
jnxJsPolicyStatsSessionDeleted	jnxJsPolicyStatsEntry 12	Number of sessions deleted from this policy.
jnxJsPolicyStatsLookups	jnxJsPolicyStatsEntry 13	Number of policy lookups performed.
jnxJsPolicyStatsCountAlarm	jnxJsPolicyStatsEntry 14	Number of alarms counted when the traffic exceeds a certain threshold configuration.

Related Documentation

- Policy Objects MIB on page 507
- Security Policy Table on page 508

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 41

Interpreting the Enterprise-Specific Reverse-Path-Forwarding MIB

- Reverse Path Forwarding MIB on page 513
- jnxRpfStatsTable on page 513

Reverse Path Forwarding MIB

The Juniper Networks enterprise-specific Reverse-Path-Forwarding (RPF) MIB monitors statistics for traffic that is rejected because of RPF processing. The RPF MIB includes one main object, **jnxRpfStats**, with an object identifier of **{jnxRpf 1}**. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-rpf.txt .

For information about the enterprise-specific Reverse Path Forwarding MIB objects, see the following topic:

- jnxRpfStatsTable on page 513

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxRpfStatsTable

The **jnxRpfStatsTable**, whose object identifier is **{jnxRpfStats 1}**, provides a list of RPF entries in table format.

The **jnxRpfStatsEntry**, whose object identifier is **{jnxRpfStatsTable 1}**, has four objects, which are listed in Table 185 on page 513.

Table 185: jnxRpfStatsEntry

Object	Object Identifier	Description
jnxRpfStatsIfIndex	**jnxRpfStatsEntry 1**	The ingress interface for traffic that is counted in an **RpfStats** entry.

Table 185: jnxRpfStatsEntry *(continued)*

Object	Object Identifier	Description
jnxRpfStatsAddrFamily	**jnxRpfStatsEntry 2**	The address family of an entry's traffic, which can be in IPv4 or IPv6 format.
jnxRpfStatsPackets	**jnxRpfStatsEntry 3**	The number of packets received on this interface, belonging to this address family, that have been rejected due to RPF processing.
jnxRpfStatsBytes	**jnxRpfStatsEntry 4**	The number of bytes received on this interface, belonging to this address family, that have been rejected due to RPF processing.

Related Documentation

- Reverse Path Forwarding MIB on page 513

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 42

Interpreting the Enterprise-Specific RMON Events and Alarms MIB

- RMON Events and Alarms MIB on page 515
- jnxRmonAlarmTable on page 516
- RMON Event and Alarm Traps on page 517

RMON Events and Alarms MIB

The Juniper Networks enterprise-specific Remote Monitoring (RMON) Events and Alarms MIB monitors objects on a device and warns the network system administrator if one of those values exceeds the defined range. The alarm monitors objects in this MIB and triggers an event when the condition (falling or rising threshold) is reached.

The Juniper Networks enterprise-specific extension to the standard RMON MIB augments the **alarmTable** with additional information about each alarm. Two new traps, **jnxRmonAlarmGetFailure** and **jnxRmonGetOk**, are also defined to indicate when problems are encountered with an alarm.

To view a complete copy of the enterprise-specific extensions to the RMON MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-rmon.txt .

For more information about RMON alarms and events, see the *Junos OS Network Management Configuration Guide*.

For information about the enterprise-specific RMON Events and Alarms MIB objects, see the following topics:

- jnxRmonAlarmTable on page 516
- RMON Event and Alarm Traps on page 517

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxRmonAlarmTable

The entries in the **jnxRmonAlarmTable**, whose object identifier is **{jnxMibs 13}**, are represented by **jnxRmonAlarmEntry**, whose object identifier is **{jnxRmonAlarmTable 1}** and are listed in Table 186 on page 516.

Table 186: jnxRmonAlarmEntry

Object	Object Identifier	Description
jnxRmonAlarmGetFailCnt	jnxRmonAlarmEntry 1	Represents the number of times the internal **Get** request for the variable monitored by this entry has failed.
jnxRmonAlarmGetFailTime	jnxRmonAlarmEntry 2	Represents the value of **sysUpTime** when an internal **Get** request for the variable monitored by this entry last failed.
jnxRmonAlarmGetFailReason	jnxRmonAlarmEntry 3	Represents the reason an internal **Get** request for the variable monitored by this entry last failed. This object contains the following values: • **other (1)**—An error was encountered that does not fit into one of the currently defined categories. • **noError (2)**—**Get** request processed successfully. • **noSuchObject (3)**—Requested object not available. • **outOfView (4)**—Requested object instance out of MIB view. • **noSuchInstance (5)**—Requested object instance not available. • **badReqId (6)**—Unexpected request ID encountered while processing **Get** request. • **oidMatchErr (7)**—Unexpected object ID encountered while processing **Get** request. • **oidBindErr (8)**—Unable to bind object ID to **Get** request PDU. • **createPktErr (9)**—Unable to create **Get** request PDU. • **badObjType (10)**—Unexpected object type encountered while processing **Get** request.
jnxRmonAlarmGetOkTime	jnxRmonAlarmEntry 4	Represents the value of **sysUpTime** when an internal **Get** request for the variable monitored by this entry succeeded and the entry left the **getFailure** state.

Copyright © 2012, Juniper Networks, Inc.

Table 186: jnxRmonAlarmEntry *(continued)*

Object	Object Identifier	Description
jnxRmonAlarmState	**jnxRmonAlarmEntry 5**	Represents the current state of this RMON alarm entry. This object contains the following values: • **unknown (1)**—Alarm entry unknown • **underCreation (2)**—Alarm entry not activated • **active (3)**—Alarm entry active and within thresholds • **startup (4)**—Alarm entry still waiting for first value • **risingThreshold (5)**—Alarm entry has crossed the rising threshold. • **fallingThreshold (6)**—Alarm entry has crossed the falling threshold • **getFailure (7)**—Alarm entry internal **Get** request failed.

Related Documentation

- RMON Events and Alarms MIB on page 515
- RMON Event and Alarm Traps on page 517

RMON Event and Alarm Traps

The following traps send notifications when there is a problem with RMON alarm processing and are listed in Table 187 on page 517.

Table 187: RMON Event and Alarm Traps

Trap	Object Identifier	Description
jnxRmonAlarmGetFailure	**jnxRmonTrapPrefix 1**	Generated when the **Get** request for an alarm variable returns an error. The specific error is identified by **jnxRmonAlarmGetFailReason**.
jnxRmonGetOk	**jnxRmonTrapPrefix 2**	Generated when the **Get** request for an alarm variable is successful. This trap is sent only if the previous **Get** requests were unsuccessful.

Related Documentation

- RMON Events and Alarms MIB on page 515
- jnxRmonAlarmTable on page 516
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 43

Interpreting the Enterprise-Specific Security Interface Extension Objects MIB

- Security Interface Extension Objects MIB on page 519
- jnxJsIfMonTable on page 519

Security Interface Extension Objects MIB

The Juniper Networks enterprise-specific Security Interface Extension Objects MIB, **jnxJsIfMIB**, whose object ID is **{jnxJsIf 1}**, defines the objects that are used to monitor the entries in the interfaces that pertain to the security management of the interface.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-if-ext.txt .

For information about the enterprise-specific Security Interface Extension Objects MIB, see the following topic:

- jnxJsIfMonTable on page 519

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsIfMonTable

jnxJsIfMonTable, whose object ID is **{jnxJsIfExtension 1}**, extends the interface entries on a particular interface to support the security related-objects listed in Table 188 on page 520. The table is indexed by **ifIndex**.

Table 188: jnxJsIfMonTable

Object	Object ID	Description
jnxJsIfMonEntry	jnxJsIfMonTable 1	Object related to interface monitoring. Sequence of parameters: • jnxJsIfMonInIcmp • jnxJsIfMonInSelf • jnxJsIfMonInVpn • jnxJsIfMonInPolicyPermit • jnxJsIfMonOutPolicyPermit • jnxJsIfMonConn • jnxJsIfMonInMcast • jnxJsIfMonOutMcast • jnxJsIfMonPolicyDeny • jnxJsIfMonNoGateParent • jnxJsIfMonTcpProxyDrop • jnxJsIfMonNoDip • jnxJsIfMonNoNspTunnel • jnxJsIfMonNoNatCon • jnxJsIfMonInvalidZone • jnxJsIfMonIpClsFail • jnxJsIfMonAuthDrop • jnxJsIfMonMultiUserAuthDrop • jnxJsIfMonLoopMultiDipDrop • jnxJsIfMonAddrSpoof • jnxJsIfMonLpDrop • jnxJsIfMonNullZone • jnxJsIfMonNoGate • jnxJsIfMonNoMinorSess • jnxJsIfMonNvecErr • jnxJsIfMonTcpSeq • jnxJsIfMonIllegalPak • jnxJsIfMonNoRoute • jnxJsIfMonAuthFail • jnxJsIfMonSaInactive • jnxJsIfMonNoSa • jnxJsIfMonSelfPktDrop
jnxJsIfMonInIcmp	jnxJsIfMonEntry 1	ICMP packets received.
jnxJsIfMonInSelf	jnxJsIfMonEntry 2	Self packets received.
jnxJsIfMonInVpn	jnxJsIfMonEntry 3	VPN packets received.
jnxJsIfMonInPolicyPermit	jnxJsIfMonEntry 4	Incoming bytes permitted by policy.

Copyright © 2012, Juniper Networks, Inc.

Table 188: jnxJsIfMonTable *(continued)*

Object	Object ID	Description
jnxJsIfMonOutPolicyPermit	jnxJsIfMonEntry 5	Outgoing bytes permitted by policy.
jnxJsIfMonConn	jnxJsIfMonEntry 6	Incoming connections established.
jnxJsIfMonInMcast	jnxJsIfMonEntry 7	Multicast packets received.
jnxJsIfMonOutMcast	jnxJsIfMonEntry 8	Multicast packets sent.
jnxJsIfMonPolicyDeny	jnxJsIfMonEntry 9	Packets dropped due to policy denial.
jnxJsIfMonNoGateParent	jnxJsIfMonEntry 10	Packets dropped due to no parent for a gate.
jnxJsIfMonTcpProxyDrop	jnxJsIfMonEntry 11	Packets dropped due to syn-attack protection.
jnxJsIfMonNoDip	jnxJsIfMonEntry 12	Packets dropped due to DIP errors.
jnxJsIfMonNoNspTunnel	jnxJsIfMonEntry 13	Packets dropped because no NSP tunnel found.
jnxJsIfMonNoNatCon	jnxJsIfMonEntry 14	Packets dropped due to no more sessions.
jnxJsIfMonInvalidZone	jnxJsIfMonEntry 15	Packets dropped because an invalid zone received the packet.
jnxJsIfMonIpClsFail	jnxJsIfMonEntry 16	Packets dropped due to IP classification failure.
jnxJsIfMonAuthDrop	jnxJsIfMonEntry 17	Packets dropped due to user authentication errors.
jnxJsIfMonMultiUserAuthDrop	jnxJsIfMonEntry 18	Packets dropped due to multiple user authentications in loopback sessions.
jnxJsIfMonLoopMultiDipDrop	jnxJsIfMonEntry 19	Packets dropped due to multiple DIP in loopback sessions.
jnxJsIfMonAddrSpoof	jnxJsIfMonEntry 20	Packets dropped due to address spoofing.
jnxJsIfMonLpDrop	jnxJsIfMonEntry 21	Packets dropped due to no loopback.
jnxJsIfMonNullZone	jnxJsIfMonEntry 22	Packets dropped due to no zone or NULL zone binding.
jnxJsIfMonNoGate	jnxJsIfMonEntry 23	Packets dropped due to no NAT gateway.

Copyright © 2012, Juniper Networks, Inc.

Table 188: jnxJsIfMonTable *(continued)*

Object	Object ID	Description
jnxJsIfMonNoMinorSess	jnxJsIfMonEntry 24	Packets dropped due to no minor session.
jnxJsIfMonNvecErr	jnxJsIfMonEntry 25	Packets dropped due to no session for gateway.
jnxJsIfMonTcpSeq	jnxJsIfMonEntry 26	Packets dropped because TCP sequence number out of window.
jnxJsIfMonIllegalPak	jnxJsIfMonEntry 27	Packets dropped because they did not make any sense.
jnxJsIfMonNoRoute	jnxJsIfMonEntry 28	Packets dropped because no route was present.
jnxJsIfMonAuthFail	jnxJsIfMonEntry 29	Packets dropped because authentication failed.
jnxJsIfMonSaInactive	jnxJsIfMonEntry 30	Packets dropped because security association (SA) is not active.
jnxJsIfMonNoSa	jnxJsIfMonEntry 31	Packets dropped because no SA found for incoming security parameter index (SPI).
jnxJsIfMonSelfPktDrop	jnxJsIfMonEntry 32	Packets dropped because there was no interest in self packets.

Related Documentation

- Security Interface Extension Objects MIB on page 519

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 44

Interpreting the Enterprise-Specific Security Screening Objects MIB

- Security Screening Objects MIB on page 523
- jnxJsScreenMonTable on page 524
- jnxJsScreenSweepTable on page 534
- jnxJsScreenNotifications on page 535
- jnxJsScreenTrapVars on page 535

Security Screening Objects MIB

The Juniper Networks enterprise-specific Security Screening Objects MIB, **jnxJsScreenMIB**, whose object ID is **{jnxJsScreening 1}**, defines the MIB for the Juniper Networks enterprise firewall screening functionality.

The Juniper Networks security firewall provides various detection methods and defense mechanisms to combat exploits at all stages of the path of execution, including:

- Screen option setting
- Firewall Denial-of-Service (DoS) attack
- Network DoS attack
- OS-specific DoS attack
- Fragment reassembly

Related Security Screening Objects MIBs include:

- **jnxJsScreenNotifications {jnxJsScreenMIB 0}**
- **jnxJsScreenObjects {jnxJsScreenMIB 1}**
- **jnxJsScreenTrapVars {jnxJsScreenMIB 2}**

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-screening.txt .

For information about the enterprise-specific Security Screening Objects MIB, see the following topics:

- jnxJsScreenMonTable on page 524
- jnxJsScreenSweepTable on page 534
- jnxJsScreenNotifications on page 535
- jnxJsScreenTrapVars on page 535

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsScreenMonTable

The **jnxJsScreenMonTable**, whose object ID is **{jnxJsScreenObjects 1}**, collects the screen attributes that monitor the various attacks to enable the Juniper Networks Security Firewall to provide deep inspection (DI) protection on each of the security device's physical interfaces. These attributes are listed in Table 207 on page 556.

The screen options can be enabled at a security zone bounded to an interface or interfaces. When these options apply to traffic reaching the security device through interfaces (via a zone), they offer protection against a malicious information gathering probe or an attack to compromise, disable, or harm a network or network resources.

Table 189: jnxJsScreenMonTable

Object	Object ID	Description
jnxJsScreenMonEntry	**jnxJsScreenMonTable 1**	The screen option monitoring statistics entry. Each entry is uniquely identified by the zone name. The data is collected on a per zone basis. There can be multiple interfaces bound to a particular zone. Hence, the statistics are aggregated across the interfaces on a per zone basis. Sequence of parameters: • **jnxJsScreenZoneName** • **jnxJsScreenNumOfIf** • **jnxJsScreenMonSynAttk** • **jnxJsScreenMonTearDrop** • **jnxJsScreenMonSrcRoute** • **jnxJsScreenMonPingDeath** • **jnxJsScreenMonAddrSpoof** • **jnxJsScreenMonLand** • **jnxJsScreenMonIcmpFlood** • **jnxJsScreenMonUdpFlood** • **jnxJsScreenMonWinnuke** • **jnxJsScreenMonPortScan** • **jnxJsScreenMonIpSweep**

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		• **jnxJsScreenMonSynFrag** • **jnxJsScreenMonTcpNoFlag** • **jnxJsScreenMonIpUnknownProt** • **jnxJsScreenMonIpOptBad** • **jnxJsScreenMonIpOptRecRt**—Record route option • **jnxJsScreenMonIpOptTimestamp**—Timestamp option • **jnxJsScreenMonIpOptSecurity** • **jnxJsScreenMonIpOptLSR**—Loose source route • **jnxJsScreenMonIpOptSSR**—Strict source route • **jnxJsScreenMonIpOptStream**—Stream options • **jnxJsScreenMonIcmpFrag** • **jnxJsScreenMonIcmpLarge** • **jnxJsScreenMonTcpSynFin** • **jnxJsScreenMonTcpFinNoAck** • **jnxJsScreenMonLimitSessSrc**—Session limit (source IP-based)
		• **jnxJsScreenMonLimitSessDest**—Session limit (destination IP-based) • **jnxJsScreenMonSynAckAck** • **jnxJsScreenMonIpFrag** • **jnxJsScreenSynAttackThresh**—Threshold data • **jnxJsScreenSynAttackTimeout**—Threshold data • **jnxJsScreenSynAttackAlmTh**—Threshold data • **jnxJsScreenSynAttackQueSize**—Threshold data • **jnxJsScreenSynAttackAgeTime**—Threshold data (obsolete in this release) • **jnxJsScreenIcmpFloodThresh**—Threshold data • **jnxJsScreenUdpFloodThresh**—Threshold data • **jnxJsScreenPortScanThresh**—Threshold data • **jnxJsScreenIpSweepThresh**—Threshold data • **jnxJsScreenSynAckAckThres**—Threshold data
jnxJsScreenZoneName	**jnxJsScreenMonEntry 1**	Name of the security zone under which the statistics are collected.
jnxJsScreenNumOfIf	**jnxJsScreenMonEntry 2**	Number of interfaces bound to this zone. Each counter contains the aggregated data of all the interfaces.
jnxJsScreenMonSynAttk	**jnxJsScreenMonEntry 3**	Number of SYN (TCP connection request) attacks. A SYN attack is a common denial of service (DoS) technique characterized by the following pattern: • Using a spoofed IP address not in use on the Internet, an attacker sends multiple SYN packets to the target machine. • For each SYN packet received, the target machine allocates resources and sends an acknowledgment (SYN-ACK) to the

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		source IP address. This can cause the target machine to allocate resources for more than 3 minutes to respond to just one SYN attack, subsequently wasting resources.
jnxJsScreenMonTearDrop	jnxJsScreenMonEntry 4	Number of teardrop attacks. Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position of the data contained in a fragmented packet relative to the data of the original unfragmented packet. When the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap. The server attempting to reassemble the packet can crash, especially if it is running an older operating system that has this vulnerability. When this option is enabled, the security device detects this discrepancy in a fragmented packet and drops it, and counts the number of packets dropped.
jnxJsScreenMonSrcRoute	jnxJsScreenMonEntry 5	Number of either loose source route option packets or strict source route attack packets. IP source route options can be used to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.
jnxJsScreenMonPingDeath	jnxJsScreenMonEntry 6	Number of ping-of-death attack packets. The maximum allowable IP packet size is 65,535 bytes, including the packet header (typically 20 bytes long). An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes. Many ping implementations, however, allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions, such as DoS, crashing, freezing, and rebooting. When the ping-of-death option is enabled, the security device detects and rejects such oversized and irregular packet sizes, even when the attacker hides the total packet size by purposefully fragmenting it.
jnxJsScreenMonAddrSpoof	jnxJsScreenMonEntry 7	Number of address spoofing attack packets. One method to gain access to a restricted network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. The mechanism to detect IP spoofing relies on route table entries.

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		For example, if a packet with source IP address 10.1.1.6 arrives at port **eth3**, but the security device has a route to 10.1.1.0/24 through port **eth1**, IP spoofing checking notes that this address arrived at an invalid interface as defined in the route table. A valid packet from 10.1.1.6 can arrive only through **eth1**, not **eth3**. The security device concludes that the packet has a spoofed source IP address and discards it.
jnxJsScreenMonLand	**jnxJsScreenMonEntry 8**	Number of land attack packets. A SYN attack combined with an IP spoof is referred to as a land attack. A land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address. The receiving victim responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the victim, causing a DoS.
jnxJsScreenMonIcmpFlood	**jnxJsScreenMonEntry 9**	Number of ICMP flood attack packets. An ICMP flood typically occurs when ICMP echo requests overload a victim with so many requests that the victim expends all its resources responding to the ICMP echo requests until it can no longer process valid network traffic. With ICMP flood protection enabled and a threshold set, if the threshold is exceeded, the victim invokes the flood attack protection feature. The default threshold value is 1000 pps. If the threshold is exceeded, the security device ignores further ICMP echo requests for the remainder of that second plus the next second as well.
jnxJsScreenMonUdpFlood	**jnxJsScreenMonEntry 10**	Number of UDP flood attack packets. UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. With UDP flood protection enabled, a threshold can be set so that when the threshold is exceeded, the system invokes UDP flood attack protection. The default threshold value is 1000 pps. If the number of UDP datagrams from one or more sources to a single destination exceeds this threshold, the security device ignores further UDP datagrams to that destination for the remainder of that second plus the next second as well.
jnxJsScreenMonWinnuke	**jnxJsScreenMonEntry 11**	Number of NetBIOS attacks. WinNuke is a DoS attack targeting any computer on the Internet running Microsoft Windows. The attacker sends a TCP segment, usually to NetBIOS port 139 of a host with an established connection with segment's urgent (URG) flag set. This practice introduces a NetBIOS fragment overlap, which causes many machines running Microsoft Windows to crash.

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
jnxJsScreenMonPortScan	jnxJsScreenMonEntry 12	Number of port scan attempt attack packets. A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval. The purpose of this attack is to scan the available services in the hope that at least one port will respond, thus identifying a service of the target. The security device should internally log the number of different ports scanned from one remote source.
jnxJsScreenMonIpSweep	jnxJsScreenMonEntry 13	Number of address sweep attempt attack packets. An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined interval. The purpose of this attack is to send ICMP packets, typically echo requests, to various hosts in the hope that at least one replies, thus uncovering an address of the target. The security device internally logs the number of ICMP packets to different addresses from one remote source.
jnxJsScreenMonSynFrag	jnxJsScreenMonEntry 14	Number of SYN fragments. IP encapsulates a TCP SYN segment in the IP packet that initiates a TCP connection. The purpose is to initiate a connection and to invoke a SYN/ACK segment response. The SYN segment typically does not contain any data since the IP packet is small and there is no legitimate reason for it to be fragmented. A fragmented SYN packet is anomalous and is suspicious. To be cautious, it might be helpful to block such fragments from entering the protected network. When the SYN fragmentation check is enabled, the security device detects and drops the packets when the IP header indicates that the packet has been fragmented while the SYN flag is set in the TCP header.
jnxJsScreenMonTcpNoFlag	jnxJsScreenMonEntry 15	Number of TCP packets with no flag set. A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Operating systems respond to such anomalies in different ways. The response, or even lack of response, from the targeted device can provide a clue as to the target's OS type. When this option is enabled, if the security device discovers such a header with a missing or malformed flags field, it drops the packet.
jnxJsScreenMonIpUnknownProt	jnxJsScreenMonEntry 16	Number of of unknown protocol IP packets. According to RFC 1700, some protocol types in an IP header are reserved and unassigned at this time. Precisely because these protocols are undefined, there is no way to know in advance whether a particular unknown protocol is benign or malicious. Unless your network makes use of a nonstandard protocol with a reserved or unassigned protocol number, a cautious stance is to

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		block such unknown elements from entering your protected network. When the Unknown Protocol Protection SCREEN option is enabled, the security device drops packets when the protocol field contains a protocol ID number of 137 or greater.
jnxJsScreenMonIpOptBad	jnxJsScreenMonEntry 17	Number of IP bad option packets. The IP protocol specifies a set of eight options that provide special routing controls, diagnostic tools, and security. These eight options can be used for malicious objectives. Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. The incorrect formatting is anomalous and potentially harmful to the intended recipient. When the Bad IP Option Protection SCREEN option is enabled, the security device detects and blocks packets when any IP option in the IP packet header is incorrectly formatted.
jnxJsScreenMonIpOptRecRt	jnxJsScreenMonEntry 18	Number of IP record option packets. The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. The record option is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose. When a record IP option is received, the security device flags it as a network reconnaissance attack and records the event for the ingress interface.
jnxJsScreenMonIpOptTimestamp	jnxJsScreenMonEntry 19	Number of IP timestamp option packets. The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Timestamp is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose. When a timestamp IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.
jnxJsScreenMonIpOptSecurity	jnxJsScreenMonEntry 20	Number of IP security option packets. The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		use. Security is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose. When a security IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.
jnxJsScreenMonIpOptLSR	jnxJsScreenMonEntry 21	Number of strict source route packets. Attackers can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.
jnxJsScreenMonIpOptStream	jnxJsScreenMonEntry 23	Number of IP stream option packets. The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Stream is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose. When a security IP option is received, the security device flags it as a network reconnaissance attack and records the event for the ingress interface.
jnxJsScreenMonIcmpFrag	jnxJsScreenMonEntry 24	Number of ICMP fragment packets. ICMP provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is wrong. With the ICMP Fragment Protection SCREEN option enabled, the security device should be able to block any ICMP packet with the More Fragments flag set or with an offset value indicated in the offset field.
jnxJsScreenMonIcmpLarge	jnxJsScreenMonEntry 25	Number of large ICMP packets. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is unusually large, something is wrong. For example, the Loki program uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a Loki agent. It might also indicate some other kind of malicious activity. When the the Large Size ICMP Packet Protection SCREEN option is enabled, the security device drops ICMP packets with a length greater than 1024 bytes.

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
jnxJsScreenMonTcpSynFin	jnxJsScreenMonEntry 26	Number of dropped TCP packets because SYN and FIN are both set. Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. When the blocking of TCP packets with both SYN and FIN is enabled, the security device drops the packet when it discovers such a header.
jnxJsScreenMonTcpFinNoAck	jnxJsScreenMonEntry 27	Number of TCP packets with FIN set, but without the ACK bit set. A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response and thereby discover an active host or an active port on a host. The use of TCP segments with the FIN flag set might evade detection and thereby help attackers succeed in their reconnaissance efforts.
jnxJsScreenMonLimitSessSrc	jnxJsScreenMonEntry 28	Number of session connections for a source IP address that exceeds the specified limit. Because all the virus-generated traffic originates from the same IP address (generally from an infected server), a source-based session limit ensures that the firewall can curb such excessive amounts of traffic. This amount is based on a threshold value of the number of concurrent sessions required to fill up the session table of the particular firewall. The default maximum for a source-based session limit is 128 concurrent sessions, which can be adjusted accordingly.
jnxJsScreenMonLimitSessDest	jnxJsScreenMonEntry 29	Number of session connections for the destination source IP address that exceeds the specified limit. The user can limit the number of concurrent sessions to the same destination IP address. An attacker can launch a distributed denial of service (DDoS) attack using “zombie agents.” Setting a destination-based session limit can ensure that the security device allows only an acceptable number of concurrent connection requests, no matter what the source, to reach any one host. The default maximum for the destination-based session limit is 128 concurrent sessions.
jnxJsScreenMonSynAckAck	jnxJsScreenMonEntry 30	Number of SYN ACK ACK attacks. When an authentication user initiates a Telnet or FTP connection, the user sends a SYN segment to the Telnet or FTP server. The security device intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At that point, the initial

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
		three-way handshake is complete. The security device sends a login prompt to the user. When a malicious user does not log in, but instead continues initiating SYN-ACK-ACK sessions, the firewall session table can fill up to the point at which the security device begins rejecting legitimate connection requests. When the SYN-ACK-ACK proxy protection option is enabled, after the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, the security device rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address.
jnxJsScreenMonIpFrag	**jnxJsScreenMonEntry 31**	Number of block IP fragment packets. As a packets travels, it is sometimes necessary to break the packet into smaller fragments based on the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system. When the block IP framentation flag is enabled, the security device blocks all IP packet fragments that it receives at interfaces bound to that zone.
Threshold Values		
jnxJsScreenSynAttackThresh	**jnxJsScreenMonEntry 32**	SYN attack threshold value. The number of SYN segments to the same destination address and port number per second required to activate the SYN proxying mechanism. Setting the appropriate threshold value requires a thorough knowledge of the normal traffic patterns at the site. For example, if the security device normally gets 2000 SYN segments per second, the threshold value should be set at 3000 segments per second.
jnxJsScreenSynAttackTimeout	**jnxJsScreenMonEntry 33**	SYN attack timeout value. The maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds.
jnxJsScreenSynAttackAlmTh	**jnxJsScreenMonEntry 34**	SYN attack alarm threshold value. The SYN attack alarm threshold causes an alarm to be generated when the number of proxied, half-completed TCP connection requests per second to the same destination address and port number exceeds its value.
jnxJsScreenSynAttackQueSize	**jnxJsScreenMonEntry 35**	SYN attack queue size. The number of proxied connection requests held in the proxied connection queue before the security device starts rejecting new connection requests.

Copyright © 2012, Juniper Networks, Inc.

Table 189: jnxJsScreenMonTable *(continued)*

Object	Object ID	Description
NOTE: The **jnxJsScreenSynAttackAgeTime** object is obsolete in this release.		
jnxJsScreenSynAttackAgeTime	**jnxJsScreenMonEntry 36**	SYN flood age time.
jnxJsScreenIcmpFloodThresh	**jnxJsScreenMonEntry 37**	ICMP attack alarm threshold value. The security device can impose a limit on the number of SYN segments permitted to pass through the firewall per second. The default attack threshold value is 1000. The valid threshold range is 1 through 100,000. When the threshold value is exceeded, an alarm is triggered.
jnxJsScreenUdpFloodThresh	**jnxJsScreenMonEntry 38**	UDP attack alarm threshold value. UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. The default threshold value is 1000 pps.
jnxJsScreenPortScanThresh	**jnxJsScreenMonEntry 39**	Port scan threshold value. The port scan threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1,000,000. By using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), the security device flags this occurrence as a port scan attack and rejects all further packets from the remote source for the remainder of the specified timeout period. The security device detects and drops the tenth packet that meets the port scan attack criterion.
jnxJsScreenIpSweepThresh	**jnxJsScreenMonEntry 40**	IP sweep threshold interval. The IP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1,000,000. By using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this occurrence as an address sweep attack and rejects all further ICMP echo requests from that host for the remainder of the specified threshold time period. The security device detects and drops the tenth packet that meets the address sweep attack criterion.
jnxJsScreenSynAckAckThres	**jnxJsScreenMonEntry 41**	SYN-ACK-ACK alarm threshold value.

Related Documentation

- Security Screening Objects MIB on page 523
- jnxJsScreenSweepTable on page 534

- jnxJsScreenNotifications on page 535
- jnxJsScreenTrapVars on page 535

jnxJsScreenSweepTable

The **jnxJsScreenSweepTable**, whose object ID is {**jnxJsScreenObjects 3**}, collects the screen attributes that monitor TCP/UDP sweep attacks. These attributes are listed in Table 190 on page 534.

Table 190: jnxJsScreenSweepTable

Object	Object ID	Description
jnxJsScreenSweepEntry	**jnxJsScreenSweepTable 1**	Maintains the TCP/UDP sweep thresholds and counters. Sequence of parameters: • **jnxJsScreenTcpSweepThresh** • **jnxJsScreenUdpSweepThresh** • **jnxJsScreenMonTcpSweep** • **jnxJsScreenMonUdpSweep**
jnxJsScreenTcpSweepThresh	**jnxJsScreenSweepEntry 1**	The TCP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000-1,000,000. If a remote host sends TCP packets to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this as a TCP sweep, and rejects all further new TCP connections initiated from that host for the remainder of the specified threshold period.
jnxJsScreenUdpSweepThresh	**jnxJsScreenSweepEntry 2**	The UDP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000-1,000,000. If a remote host sends UDP packets to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this as a UDP sweep, and rejects all further new UDP connections from that host for the remainder of the specified threshold period.
jnxJsScreenMonTcpSweep	**jnxJsScreenSweepEntry 3**	Number of TCP sessions dropped as a result of a TCP sweep.
jnxJsScreenMonUdpSweep	**jnxJsScreenSweepEntry 4**	Number of UDP packets dropped as a result of a UDP sweep.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Security Screening Objects MIB on page 523
- jnxJsScreenMonTable on page 524
- jnxJsScreenNotifications on page 535
- jnxJsScreenTrapVars on page 535

jnxJsScreenNotifications

The **jnxJsScreenNotifications** object, whose object ID is **{jnxJsScreenNotifications 0}**, specifies the screen notifications listed in Table 191 on page 535.

Table 191: jnxJsScreenNotifications

Object	Object ID	Description
jnxJsScreenAttack	**jnxJsScreenNotifications 1**	Specifies that the number of bytes per minute has exceeded the specified threshold. • **jnxJsScreenZoneName** — Specifies the zone name in which the attack is occurring. • **ifName** — Specifies the interface at which the attack is occurring. • **jnxJsScreenAttackType** — Specifies the type of attack. • **jnxJsScreenAttackCounter** — Displays the number of attacks recorded. The value is the aggregated statistic of all the interfaces bound to that specific zone. • **jnxJsScreenAttackDescr** — Provides a general text description of this attack or the trap.
jnxJsScreenCfgChange	**jnxJsScreenNotifications 2**	Specifies the status of the screening option. • **jnxJsScreenCfgStatus** — Specifies if the screening option is enabled or disabled.

Related Documentation

- Security Screening Objects MIB on page 523
- jnxJsScreenMonTable on page 524
- jnxJsScreenSweepTable on page 534
- jnxJsScreenTrapVars on page 535

jnxJsScreenTrapVars

The **jnxJsScreenTrapVars** object identifies the screen trap variables listed in Table 192 on page 536.

Table 192: jnxJsScreenTrapVars

Object	Object ID	Description
jnxJsScreenAttackType	**jnxJsScreenTrapVars 1**	Returns one of the following integer values to Identify the type of attack: • **32**—tcpsweeping • **33**—udpsweeping
jnxJsScreenAttackCounter	**jnxJsScreenTrapVars 2**	Returns an integer which defines the threshold that triggered the trap.
jnxJsScreenAttackDescr	**jnxJsScreenTrapVars 3**	Represents a string value that displays the description of the attack
jnxJsScreenCfgStatus	**jnxJsScreenTrapVars 4**	Returns one of the following integer values to indicate the state of the screening option: • **1**–disabled • **2**–enabled

Related Documentation

- Security Screening Objects MIB on page 523
- jnxJsScreenMonTable on page 524
- jnxJsScreenSweepTable on page 534
- jnxJsScreenNotifications on page 535

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 45

Interpreting the Enterprise-Specific SNMP IDP MIB

- SNMP IDP MIB on page 537
- jnxJsIdpAttackTable on page 537
- IDP Signature Update and Attack Traps on page 538

SNMP IDP MIB

The Juniper Networks enterprise-specific SNMP IDP MIB, **jnxJsIdpMIB**, whose object ID is **{jnxJsIdpRoot 1}**, extends SNMP support to the following features on Juniper Networks SRX100, SRX210, SRX220, SRX240, SRX550, and SRX650 Services Gateways:

- Key monitoring and threshold-crossing traps
- Attack-related monitoring and traps
- IDP database update status and traps

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-js-idp.txt .

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

- jnxJsIdpObjects
- jnxJsIdpAttackTable on page 537
- IDP Signature Update and Attack Traps on page 538

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsIdpAttackTable

The **jnxJsIdpAttackTable**, whose object ID is **{jnxJsIdpObjects 7}**, contains the details of each attack and the number of hits involved. The objects are listed in Table 193 on page 538.

Table 193: jnxJsIdpAttackTable

Object	Object ID	Description
jnxJsIdpAttackEntry	**jnxJsIdpAttackTable 1**	The attributes of an attack Sequence of parameters: • **jnxJsIdpAttackIndex** • **jnxJsIdpAttackName** • **jnxJsIdpAttackHits**
jnxJsIdpAttackIndex	**jnxJsIdpAttackEntry 1**	The index for the attack table.
jnxJsIdpAttackName	**jnxJsIdpAttackEntry 2**	The name of an attack identified by IDP.
jnxJsIdpAttackHits	**jnxJsIdpAttackEntry 3**	The number of times an attack has been launched.

Related Documentation

- SNMP IDP MIB on page 537
- IDP Signature Update and Attack Traps on page 538

IDP Signature Update and Attack Traps

The IDP MIB generates an SNMP trap whenever you install or update a signature, and provides information about the last updated version and a timestamp. It also generates SNMP traps to notify users about attacks that are detected on the device.

See Table 194 on page 538 for information about the signature update and attack traps.

Table 194: IDP Signature Update and Attack Traps

Object	Object ID	Description
jnxJsIdpSignatureUpdateNotification	**jnxJsIdpNotificationPrefix 1**	Signifies that a signature update has occurred.
jnxJsIdpAttackLogNotification	**jnxJsIdpNotificationPrefix 2**	Signifies that attacks have been detected. **jnxJsIdpAttackName** lists the names of the attacks that have been detected so far, and **jnxJsIdpAttackHits** is the number of times each attack has been detected.

Related Documentation

- SNMP IDP MIB on page 537
- jnxJsIdpObjects
- jnxJsIdpAttackTable on page 537
- Configuring SNMP Trap Options and Groups on a Device Running Junos OS on page 56
- Managing Traps and Informs

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 46

Interpreting the Enterprise-Specific Source Class Usage MIB

- Source Class Usage MIB on page 539
- jnxScuStatsTable on page 539

Source Class Usage MIB

The Juniper Networks enterprise-specific Source Class Usage (SCU) MIB counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge.

The enterprise-specific SCU MIB is an object of the **jnxMibs** branch of the enterprise-specific MIB **{enterprise 2636}** and has an object identifier of **{jnxMIB 16}**. The enterprise-specific SCU MIB includes one object, **jnxScuStats**, which has an object identifier of **{jnxScu 1}**. For information about configuring source and destination class usage, see the *Junos OS Policy Framework Configuration Guide* and the *Junos OS Network Interfaces Configuration Guide*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-scu.txt .

NOTE: **Class-based filter match conditions are not supported on J Series Services Routers.**

For information about the enterprise-specific SCU MIB objects, see the following topic:

- jnxScuStatsTable on page 539

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxScuStatsTable

The **jnxRpfStatsTable**, whose object identifier is **{jnxRpfStats 1}**, provides a list of RPF entries in table format.

The **jnxRpfStatsEntry**, whose object identifier is **{jnxRpfStatsTable 1}**, has four objects, which are listed in Table 195 on page 540.

Table 195: jnxRpfStatsEntry

Object	Object Identifier	Description
jnxRpfStatsIfIndex	**jnxRpfStatsEntry 1**	The ingress interface for traffic that is counted in an **RpfStats** entry.
jnxRpfStatsAddrFamily	**jnxRpfStatsEntry 2**	The address family of an entry's traffic, which can be in IPv4 or IPv6 format.
jnxRpfStatsPackets	**jnxRpfStatsEntry 3**	The number of packets received on this interface, belonging to this address family, that have been rejected due to RPF processing.
jnxRpfStatsBytes	**jnxRpfStatsEntry 4**	The number of bytes received on this interface, belonging to this address family, that have been rejected due to RPF processing.

Related Documentation

- Source Class Usage MIB on page 539

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 47

Interpreting the Enterprise-Specific SPU Monitoring MIB

- SPU Monitoring Objects MIB on page 541
- SPU Monitoring Objects Table on page 542
- Cluster Objects Table on page 543

SPU Monitoring Objects MIB

The Juniper Networks enterprise-specific Services Processing Unit (SPU) Monitoring Objects MIB, **jnxJsSecPolicyMIB**, whose object ID is **{jnxJsSPUMonitoringMIB 1}**, defines the MIB for SPU monitoring for SRX5600 and SRX5800 services gateways.

Related MIB objects include the following:

- **jnxJsSPUMonitoringObjectsTable**, whose object identifier is **{jnxJsSPUMonitoringMIB 1}**, provides statistics on the utilization of SPUs.
- **jnxJsSPUMonitoringCurrentTotalSession**, whose object identifier is **{jnxJsSPUMonitoringMIB 2}**, provides the total number of sessions in use at the system level.
- **jnxJsSPUMonitoringMaxTotalSession**, whose object identifier is **{jnxJsSPUMonitoringMIB 3}**, provides the maximum number of sessions possible at the system level.
- **jnxSPUClusterObjectsTable**, whose object identifier is **{jnxJsSPUMonitoringMIB 4}**, identifies the SPU monitoring objects in a chassis cluster.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt

For information about the enterprise-specific SPU Monitoring Objects MIB or SPU Cluster MIB, see the following topics:

- SPU Monitoring Objects Table on page 542
- Cluster Objects Table on page 543

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

SPU Monitoring Objects Table

The object identifier for the **jnxJsMonitoringObjectsTable** is **{jnxJsSPUMonitoringMIB 1}**. SPU monitoring objects provide statistical information related to utilization of SPUs. Table 196 on page 542 describes the SPU monitoring objects.

Table 196: SPU Monitoring Objects Table

Object	Object ID	Description
jnxJsSPUMonitoringObjects	jnxJsSPUMonitoringObjectsTable 1	Each entry collects information about the memory utilization for an SPU.
JnxJsSPUMonitoringObjectsEntry		Indexes: • jnxJsSPUMonitoringIndex • jnxJsSPUMonitoringFPCIndex • jnxJsSPUMonitoringSPUIndex Sequence of parameters: • jnxJsSPUMonitoringCPUUsage • jnxJsSPUMonitoringMemoryUsage • jnxJsSPUMonitoringCurrentFlowSession • jnxJsSPUMonitoringMaxFlowSession • jnxJsSPUMonitoringCurrentCPSession • jnxJsSPUMonitoringMaxCPSession • jnxJsSPUMonitoringMaxTotalSession • jnxJsSPUMonitoringNodeIndex • jnxJsSPUMonitoringNodeDescr
jnxJsSPUMonitoringIndex	jnxJsSPUMonitoringObjectsEntry 1	Indicates an SPU's overall index in the system.
jnxJsSPUMonitoringFPCIndex	jnxJsSPUMonitoringObjectsEntry 2	Indicates which FPC the SPU is on .
jnxJsSPUMonitoringSPUIndex	jnxJsSPUMonitoringObjectsEntry 3	Indicates the index of an SPU inside the FPC.
jnxJsSPUMonitoringCPUUsage	jnxJsSPUMonitoringObjectsEntry 4	Indicates the current utilization percentage of an SPU.
jnxJsSPUMonitoringMemoryUsage	jnxJsSPUMonitoringObjectsEntry 5	Indicates the current percentage of memory usage of an SPU.
jnxJsSPUMonitoringCurrentFlowSession	jnxJsSPUMonitoringObjectsEntry 6	Indicates the current flow sessions of an SPU.

Copyright © 2012, Juniper Networks, Inc.

Table 196: SPU Monitoring Objects Table *(continued)*

Object	Object ID	Description
jnxJsSPUMonitoringMaxFlowSession	jnxJsSPUMonitoringObjectsEntry 7	Indicates the maximum flow sessions of an SPU.
jnxJsSPUMonitoringCurrentCPSession	jnxJsSPUMonitoringObjectsEntry 8	Indicates the current number of central point (CP) sessions on an SPU.
jnxJsSPUMonitoringMaxCPSession	jnxJsSPUMonitoringObjectsEntry 9	Indicates the maximum number of CP sessions on an SPU.
jnxJsSPUMonitoringMaxTotalSession	jnxJsSPUMonitoringObjectsEntry 10	Indicates the maximum number of sessions on the device.
jnxJsSPUMonitoringNodeIndex	jnxJsSPUMonitoringObjectsEntry 11	Identifies a chassis in single or cluster mode.
jnxJsSPUMonitoringNodeDescr	jnxJsSPUMonitoringObjectsEntry 12	Describes the chassis cluster node information.

Related Documentation

- SPU Monitoring Objects MIB on page 541

Cluster Objects Table

The object identifier for the **jnxJsClusterObjectsTable** is **{jnxJsSPUMonitoringMIB 4}**. The cluster objects provide statistical information for each node. Table 197 on page 543 describes the cluster objects.

Table 197: Cluster Objects Table

Object	Object ID	Description
jnxJsClusterObjectsEntry	jnxJsSPUMonitoringObjectsTable 1	Each entry collects information per node.
JnxJsClusterObjectsEntry	–	Index: • jnxJsClusterMonitoringNodeIndex Sequence of parameters: • jnxJsClusterMonitoringNodeDescr • jnxJsNodeCurrentTotalSession • jnxJsNodeMaxTotalSession • jnxJsNodeSessionCreationPerSecond
jnxJsClusterMonitoringIndex	jnxJsClusterObjectsEntry 1	Indicates the index number of the node.
jnxJsClusterMonitoringNodeDescr	jnxJsClusterObjectsEntry 2	Describes the node being monitored.

Table 197: Cluster Objects Table *(continued)*

Object	Object ID	Description
jnxJsNodeCurrentTotalSession	jnxJsClusterObjectsEntry 3	Indicates the total sessions currently running on the node.
jnxJsNodeMaxTotalSession	jnxJsClusterObjectsEntry 4	Indicates the maximum number of sessions that can run on the node.
jnxJsNodeSessionCreationPerSecond	jnxJsClusterObjectsEntry 5	Indicates the number of sessions created per second on the node.

Related Documentation

- SPU Monitoring Objects MIB on page 541

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 48

Interpreting the Enterprise-Specific System Log MIB

- System Log MIB on page 545
- jnxSyslogTable on page 546
- jnxSyslogAvTable on page 548

System Log MIB

Event policies can include an action that raises traps for events based on system log messages. This feature enables notification of an SNMP trap-based application when an important system log message occurs. You can convert any system log message (for which there are no corresponding traps) into a trap. This feature is valuable for customers who use network management system traps rather than system log messages to monitor their networks. For more information about converting system log messages into traps, see the *Junos OS Configuration and Operations Automation Guide*.

The Juniper Networks enterprise-specific System Log MIB, whose object identifier is **{jnxMibs 35}**, provides support for this feature.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-syslog.txt .

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

- jnxSyslogTable on page 546
- jnxSyslogAvTable on page 548

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxSyslogTable

The **jnxSyslogTable**, whose object identifier is **{jnxSyslog 1}**, provides information about each system log message generated by the router.

The **jnxSyslogEntry**, whose object identifier is **{jnxSyslogTable 1}**, has nine objects, which are listed in Table 198 on page 546. Each entry provides information about a single system log message.

Table 198: jnxSyslogTable

Object	Object Identifier	Description
jnxSyslogID	**jnxSyslogEntry 1**	System log message identifier. This identifier is a numerical value and cannot be consecutive. This identifier is also used as the primary index in **jnxSyslogAttrValTable**.
jnxSyslogEventName	**jnxSyslogEntry 2**	An octet string that contains the system log event name.
jnxSyslogTimestamp	**jnxSyslogEntry 3**	Time the message was generated. This value is expressed as **DateAndTime**.
jnxSyslogSeverity	**jnxSyslogEntry 4**	Severity of the system log message. The severity value is equal to the value that the system log uses + 1. For example, an emergency message (severity 0 in the system log) has a severity of 1. Severity values include: • **emergency (1)** • **alert (2)** • **critical (3)** • **error (4)** • **warning (5)** • **notice (6)** • **info (7)** • **debug (8)**
jnxSyslogFacility	**jnxSyslogEntry 5**	Identifies the facility used to generate the log message. (See Table 199 on page 547.)
jnxSyslogProcessID	**jnxSyslogEntry 6**	Process ID of the process that generated the system log message.
jnxSyslogProcessName	**jnxSyslogEntry 7**	Process that generated the system log message.
jnxSyslogHostName	**jnxSyslogEntry 8**	Hostname of the machine that generated the system log.
jnxSyslogMessage	**jnxSyslogEntry 9**	System log message that was generated.

Copyright © 2012, Juniper Networks, Inc.

Table 199 on page 547 lists the facilities that generate system log messages.

Table 199: Facilities That Generate System Log Messages

Index	Item	Description
1	kernel	Kernel messages
2	user	User level messages
3	mail	Mail system
4	daemon	System processes
5	auth	Authorization messages
6	syslog	Messages generated by the system log process (**syslogd**)
7	lpr	Line printer subsystem
8	news	Network news subsystem
9	uucp	UUCP subsystem
10	cron	Clock process
11	authPriv	Authorization messages
12	ftp	FTP process
13	ntp	NTP subsystem
14	security	Security subsystems (for example, firewall)
15	console	**/dev/console** output
16	reserved	Reserved for system use
17	local0	–
18	dfc	Junos names
19	local2	–
20	firewall	Junos names
21	pfe	Junos names
22	conflict	Junos names
23	change	Junos names

Copyright © 2012, Juniper Networks, Inc.

Table 199: Facilities That Generate System Log Messages *(continued)*

Index	Item	Description
24	interact	Junos names

Related Documentation

- System Log MIB on page 545
- jnxSyslogAvTable on page 548

jnxSyslogAvTable

The **jnxSyslogAvTable**, whose object identifier is **{jnxSyslogNotifyVars 2}**, provides information about each system log message generated by the router.

The **jnxSyslogAvEntry**, whose object identifier is **{jnxSyslogAvTable 1}**, has three objects, which are listed in Table 200 on page 548. Each entry provides information about attribute value pairs of system log messages generated by a device.

Table 200: jnxSyslogAvTable

Object	Object Identifier	Description
jnxSyslogAvIndex	**jnxSyslogAvEntry 1**	Index for the attribute value pair in the system log message.
jnxSyslogAvAttribute	**jnxSyslogAvEntry 2**	Attribute of the system log message (identified by **jnxSyslogID**).
jnxSyslogAvValue	**jnxSyslogAvEntry 3**	Value of the attribute (identified by **jnxSyslogAvAttribute**).

Related Documentation

- System Log MIB on page 545
- jnxSyslogTable on page 546

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 49

Interpreting the Enterprise-Specific Traceroute MIB

- Traceroute MIB on page 549
- jnxTraceRouteCtlTable on page 549

Traceroute MIB

The Juniper Networks enterprise-specific Traceroute MIB supports Junos OS extensions of traceroutes and remote operations. Items in this MIB are created when entries are created in the **traceRouteCtlTable** of the Traceroute MIB. Each item is indexed exactly the same way as it is in the enterprise-specific Traceroute MIB. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-traceroute.txt .

For more information about using the Traceroute MIB and enterprise-specific Traceroute MIB, see the *Junos OS Network Management Configuration Guide*.

For information about the enterprise-specific Traceroute MIB objects, see the following topic:

- jnxTraceRouteCtlTable on page 549

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxTraceRouteCtlTable

The **jnxTraceRouteCtlTable**, whose object identifier is **{jnxTraceRouteObjects 2}**, defines the **jnxTraceRoute** control table for providing enterprise-specific options to the corresponding **traceRouteCtlEntry**.

Each **jnxTraceRouteCtlEntry** has two indexes that are identical to those of the corresponding **TraceRouteCtlEntry**. Entries created in **TraceRouteCtlTable** are mirrored here and are listed in Table 201 on page 550.

Copyright © 2012, Juniper Networks, Inc.

Table 201: jnxTraceRouteCtlTable

Object	Object Identifier	Description
jnxTRCtlOwnerIndex	**jnxTraceRouteCtlEntry 1**	Specifies the first index. It is identical to **jnxTraceRouteCtlOwnerIndex** of the corresponding **jnxTraceRouteCtlEntry** in the **jnxTraceRouteCtlTable**.
jnxTRCtlTestName	**jnxTraceRouteCtlEntry 2**	Specifies the other index. It is identical to **jnxTraceRouteCtlTestName** of the corresponding **jnxTraceRouteCtlEntry** in the **jnxTraceRouteCtlTable**.
jnxTRCtlIfName	**jnxTraceRouteCtlEntry 3**	Specifies the name of the outgoing interface for traceroute probes. This is the name-based complement to **traceRouteCtlIfIndex**. A zero-length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: • **traceRouteCtlIfIndex** (see **traceRouteCtlTable** in the "Traceroute MIB" on page 549) • **jnxTRCtlIfName** • **jnxTRCRoutingInstanceName**
jnxTRCtlRoutingInstanceName	**jnxTraceRouteCtlEntry 4**	Specifies the name of the routing instance used when directing outgoing traceroute packets. The instance name specified must be configured at the **[edit routing-instances]** hierarchy level of the Junos OS configuration.

Related Documentation

- Traceroute MIB on page 549

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 50

Interpreting the Enterprise-Specific Utility MIB

- Utility MIB on page 551
- jnxUtilCounter32Table on page 552
- jnxUtilCounter64Table on page 552
- jnxUtilIntegerTable on page 553
- jnxUtilUintTable on page 553
- jnxUtilStringTable on page 554

Utility MIB

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

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

Each data type has an arbitrary ASCII name, which is defined when the data is populated, and a timestamp that shows the last time when the data instance was modified. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/ reference/mibs/mib-jnx-util.txt .

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

- jnxUtilCounter32Table on page 552
- jnxUtilCounter64Table on page 552
- jnxUtilIntegerTable on page 553

Copyright © 2012, Juniper Networks, Inc.

- jnxUtilUintTable on page 553
- jnxUtilStringTable on page 554

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxUtilCounter32Table

jnxUtilCounter32Table, whose object ID is **{jnxUtilData 1}**,contains information about 32-bit counters.

Each **jnxUtilCounter32Entry** has the objects listed in Table 202 on page 552.

Table 202: jnxUtilCounter32Entry

Object	Object ID	Description
jnxUtilCounter32Name	jnxUtilCounter32Entry 1	Name assigned to the instance.
jnxUtilCounter32Value	jnxUtilCounter32Entry 2	Value of the instance.
jnxUtilCounter32Time	jnxUtilCounter32Entry 3	Time when the instance was last populated.

Related Documentation

- Utility MIB on page 551
- jnxUtilCounter64Table on page 552
- jnxUtilIntegerTable on page 553
- jnxUtilUintTable on page 553
- jnxUtilStringTable on page 554

jnxUtilCounter64Table

jnxUtilCounter64Table, whose object ID is **{jnxUtilData 2}**, contains information about 64-bit counters.

Each **jnxUtilCounter64Entry** has the objects listed in Table 203 on page 552.

Table 203: jnxUtilCounter64Entry

Object	Object ID	Description
jnxUtilCounter64Name	jnxUtilCounter64Entry 1	Name assigned to the instance.
jnxUtilCounter64Value	jnxUtilCounter64Entry 2	Value of the instance.
jnxUtilCounter64Time	jnxUtilCounter64Entry 3	Time when the instance was last populated.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Utility MIB on page 551
- jnxUtilCounter32Table on page 552
- jnxUtilIntegerTable on page 553
- jnxUtilUintTable on page 553
- jnxUtilStringTable on page 554

jnxUtilIntegerTable

jnxUtilIntegerTable, whose object ID is **{jnxUtilData 3}**, contains information about signed integer values.

Each **jnxUtilIntegerEntry** contains the objects listed in Table 204 on page 553.

Table 204: jnxUtilIntegerEntry

Object	Object ID	Description
jnxUtilIntegerName	jnxUtilIntegerEntry 1	Name assigned to the instance.
jnxUtilIntegerValue	jnxUtilIntegerEntry 2	Value of the instance.
jnxUtilIntegerTime	jnxUtilIntegerEntry 3	Time when the instance was last populated.

Related Documentation

- Utility MIB on page 551
- jnxUtilCounter32Table on page 552
- jnxUtilCounter64Table on page 552
- jnxUtilUintTable on page 553
- jnxUtilStringTable on page 554

jnxUtilUintTable

jnxUtilUintTable, whose object ID is **{jnxUtilData 4}**, contains information about unsigned integer values.

Each **jnxUtilUintEntry** has the objects listed in Table 205 on page 553.

Table 205: jnxUtilUintEntry

Object	Object ID	Description
jnxUtilUintName	jnxUtilUintEntry 1	Name assigned to the instance.
jnxUtilUintValue	jnxUtilUintEntry 2	Value of the instance.
jnxUtilUintTime	jnxUtilUintEntry 3	Time when the instance was last populated.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Utility MIB on page 551
- jnxUtilCounter32Table on page 552
- jnxUtilCounter64Table on page 552
- jnxUtilIntegerTable on page 553
- jnxUtilStringTable on page 554

jnxUtilStringTable

jnxUtilStringTable, whose object ID is **{jnxUtilData 5}**, contains information about octet strings.

Each **jnxUtilStringEntry** contains the objects listed in Table 206 on page 554.

Table 206: jnxUtilStringEntry

Object	Object ID	Description
jnxUtilStringName	**jnxUtilStringEntry 1**	Name assigned to the instance.
jnxUtilStringValue	**jnxUtilStringEntry 2**	Value of the instance.
jnxUtilStringTime	**jnxUtilStringEntry 31**	Time when the instance was last populated.

Related Documentation

- Utility MIB on page 551
- jnxUtilCounter32Table on page 552
- jnxUtilCounter64Table on page 552
- jnxUtilIntegerTable on page 553
- jnxUtilUintTable on page 553

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 51

Interpreting the VPN Certificate Objects MIB

- VPN Certificate Objects MIB on page 555
- jnxJsLoadedCaCertTable on page 555
- jnxJsLoadedLocalCertTable on page 556

VPN Certificate Objects MIB

The Juniper Networks enterprise-specific VPN Certificate Objects MIB, **jnxJsCertificateMIB**, whose object ID is **{jnxJsCertificates 1}**, defines the objects that are used to monitor reference and attributes to the certificates.

A related VPN Certificate Object MIB is **jnxJsCertificateObjects {jnxJsCertificateMIB 1}**.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos12.1/topics/reference/mibs/mib-jnx-js-cert.txt .

For information about the enterprise-specific VPN Certificate Objects MIB, see the following topics:

- jnxJsLoadedCaCertTable on page 555
- jnxJsLoadedLocalCertTable on page 556

Related Documentation

- Juniper Networks Enterprise-Specific MIBs on page 31
- Standard SNMP MIBs Supported by Junos OS on page 17

jnxJsLoadedCaCertTable

The **jnxJsLoadedCaCertTable**, whose object ID is **{jnxJsCertificateObjects 1}**, exposes the loaded Certification Authority (CA)-issued certificate objects listed in Table 207 on page 556. Certificates are used when establishing a secure connection in the device.

Table 207: jnxJsLoadedCaCertTable

Object	Object ID	Description
jnxJsLoadedCaCertEntry	jnxJsLoadedCaCertTable 1	Loaded CA certificate entry. The loaded CA certificate entry is indexed by the CA certification name. Sequence of parameters: • jnxJsLoadedCaCertName • jnxJsLoadedCaCertSubject • jnxJsLoadedCaCertExpire • jnxJsLoadedCaCertIssuer
jnxJsLoadedCaCertName	jnxJsLoadedCaCertEntry 1	Loaded CA certificate name. This name is unique within the loaded CA certifcates.
jnxJsLoadedCaCertSubject	jnxJsLoadedCaCertEntry 2	CA certificate subject.
jnxJsLoadedCaCertExpire	jnxJsLoadedCaCertEntry 3	Expiration time and date of CA certificate.
jnxJsLoadedCaCertIssuer	jnxJsLoadedCaCertEntry 4	Issuer of CA certificate.

Related Documentation

- VPN Certificate Objects MIB on page 555
- jnxJsLoadedLocalCertTable on page 556

jnxJsLoadedLocalCertTable

The **jnxJsLoadedLocalCertTable**, whose object ID is **{jnxJsCertificateObjects 2}**, exposes the loaded local certificate objects listed in Table 208 on page 556. Certificates are used when establishing a secure connection in the device.

Table 208: jnxJsLoadedLocalCertTable

Object	Object ID	Description
jnxJsLoadedLocalCertEntry	jnxJsLoadedLocalCertTable 1	Default certificate entry. This entry is indexed by the certification name. Sequence of parameters: • jnxJsLoadedLocalCertName • jnxJsLoadedLocalCertSubject • jnxJsLoadedLocalCertExpire • jnxJsLoadedLocalCertIssuer
jnxJsLoadedLocalCertName	jnxJsLoadedLocalCertEntry 1	Name of the local certificate. The certificate name is unique within the loaded local certificates.
jnxJsLoadedLocalCertSubject	jnxJsLoadedLocalCertEntry 2	Certificate subject.
jnxJsLoadedLocalCertExpire	jnxJsLoadedLocalCertEntry 3	Expiration time and date of the local certificate.

Copyright © 2012, Juniper Networks, Inc.

Table 208: jnxJsLoadedLocalCertTable *(continued)*

Object	Object ID	Description
jnxJsLoadedLocalCertIssuer	**jnxJsLoadedLocalCertEntry 4**	Issuer of the local certificate.

Related Documentation

- VPN Certificate Objects MIB on page 555
- jnxJsLoadedCaCertTable on page 555

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

PART 4

Accounting Options

- Accounting Options Overview on page 561
- Configuring Accounting Options on page 563

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 52

Accounting Options Overview

This chapter contains the following topic:

- Accounting Options Overview on page 561

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 209 on page 561.

Table 209: 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
- Accounting Options Configuration on page 563
- Configuring Accounting-Data Log Files on page 566

- Configuring the Interface Profile on page 569
- Configuring the Filter Profile on page 572
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 53

Configuring Accounting Options

- Accounting Options Configuration on page 563
- Configuring Accounting-Data Log Files on page 566
- Configuring the Interface Profile on page 569
- Configuring the Filter Profile on page 572
- Example: Configuring a Filter Profile on page 574
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 574
- Understanding Source Class Usage and Destination Class Usage Options on page 576
- Configuring SCU or DCU on page 577
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579
- Configuring Class Usage Profiles on page 580
- Configuring the MIB Profile on page 583
- Configuring the Routing Engine Profile on page 584

Accounting Options Configuration

This topic contains the following sections:

- Accounting Options—Full Configuration on page 563
- Minimum Accounting Options Configuration on page 564

Accounting Options—Full Configuration

To configure accounting options, include the following statements at the **[edit accounting-options]** hierarchy level:

```
accounting-options {
  class-usage-profile profile-name {
    file filename;
    interval minutes;
    destination-classes {
      destination-class-name;
    }
    source-classes {
      source-class-name;
    }
    file filename {
```

```
            archive-sites {
                site-name;
            }
            files number;
            nonpersistent;
            size bytes;
            source-classes time
            transfer-interval minutes;
        }
        filter-profile profile-name {
            counters {
                counter-name;
            }
            file filename;
            interval minutes;
        }
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
    mib-profile profile-name {
        file filename;
        interval seconds;
        object-names {
            mib-object-name;
        }
        operation operation-name;
    }
    routing-engine-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
}
```

By default, accounting options are disabled.

Minimum Accounting Options Configuration

To enable accounting options on the router, you must perform at least the following tasks:

- Configure accounting options by including a **file** statement and one or more **source-class-usage**, **destination-class-profile**, **filter-profile**, **interface-profile**, **mib-profile**, or **routing-engine-profile** statements at the **[edit accounting-options]** hierarchy level:

```
[edit]
accounting-options {
    class-usage-profile profile-name {
        file filename;
```

 Copyright © 2012, Juniper Networks, Inc.

```
        interval minutes;
        source-classes {
          source-class-name;
          destination-classes {
            destination-class-name;
          }
        }
        file filename {
          archive-sites {
            site-name;
          }
          files number;
          size bytes;
          transfer-interval minutes;
        }
        filter-profile profile-name {
          counters {
            counter-name;
          }
          file filename;
          interval minutes;
        }
        interface-profile profile-name {
          fields {
            field-name;
          }
          file filename;
          interval minutes;
        }
        mib-profile profile-name {
          file filename;
          interval minutes;
          object-names {
            mib-object-name;
          }
          operation operation-name;
        }
        routing-engine-profile profile-name {
          fields {
            field-name;
          }
          file filename;
          interval minutes;
        }
      }
    }
```

- Apply the profiles to the chosen interfaces or filters.

  Apply an interface profile to a physical or logical interface by including the **accounting-profile** statement at either the **[edit interfaces *interface-name*]** or the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level. For more information about interface profiles, see the *Junos OS Network Interfaces Configuration Guide*.

```
[edit interfaces]
interface-name {
  accounting-profile profile-name;
```

```
    unit logical-unit-number {
      accounting-profile profile-name;
    }
  }
```

NOTE: You do not apply destination class profiles to interfaces. Although the interface needs to have the **destination-class-usage** statement configured, the destination class profile automatically finds all interfaces with the destination class configured.

Apply a filter profile to a firewall filter by including the **accounting-profile** statement at the **[edit firewall filter *filter-name*]** hierarchy level:

```
[edit firewall]
filter filter-name {
  accounting-profile profile-name;
}
```

You do not need to apply the Routing Engine profile to an interface because the statistics are collected on the Routing Engine itself.

Related Documentation

- Accounting Options Overview on page 561
- Understanding Device Management Functions in Junos OS
- Configuring Accounting-Data Log Files on page 566
- Configuring the Interface Profile on page 569
- Configuring the Filter Profile on page 572
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5

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.

Copyright © 2012, Juniper Networks, Inc.

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 567
- Configuring the Maximum Size of the File on page 567
- Configuring the Maximum Number of Files on page 568
- Configuring the Start Time for File Transfer on page 568
- Configuring the Transfer Interval of the File on page 568
- Configuring Archive Sites on page 569

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.

Copyright © 2012, Juniper Networks, Inc.

Configuring Archive Sites

After a file reaches its maximum size or the **transfer-interval** time is exceeded, the file is closed, renamed, and, if you configured an archive site, transferred to a remote host. To configure archive sites, include the **archive-sites** statement at the **[edit accounting-options file *filename*]** hierarchy level:

```
[edit accounting-options file filename]
archive-sites {
   site-name;
}
```

site-name is any valid FTP URL. For more information about specifying valid FTP URLs, see the *Junos OS System Basics Configuration Guide*. You can specify more than one URL, in any order. When a file is archived, the router or switch attempts to transfer the file to the first URL in the list, trying the next site in the list only if the transfer does not succeed. The log file is stored at the archive site with a filename of the format ***router-name_log-filename_timestamp***.

Related Documentation

- Accounting Options Overview on page 561
- Understanding Device Management Functions in Junos OS
- Accounting Options Configuration on page 563
- Configuring the Interface Profile on page 569
- Configuring the Filter Profile on page 572
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5

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.

Copyright © 2012, Juniper Networks, Inc.

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 *Junos OS Policy Framework Configuration Guide*.

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

- Configuring Fields on page 570
- Configuring the File Information on page 570
- Configuring the Interval on page 570
- Example: Configuring the Interface Profile on page 571

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.

Copyright © 2012, Juniper Networks, Inc.

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 561
- Understanding Device Management Functions in Junos OS
- Accounting Options Configuration on page 563
- Configuring Accounting-Data Log Files on page 566
- Configuring the Filter Profile on page 572
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5

Configuring the Filter Profile

A filter profile specifies error and statistics information collected and written to a file. A filter profile must specify counter names for which statistics are collected.

To configure a filter profile, include the **filter-profile** statement at the **[edit accounting-options]** hierarchy level:

```
[edit accounting-options]
filter-profile profile-name {
   counters {
      counter-name;
   }
   file filename;
   interval minutes;
}
```

To apply the filter profile, include the **accounting-profile** statement at the **[edit firewall filter *filter-name*]** hierarchy level. For more information about firewall filters, see the *Junos OS Network Interfaces Configuration Guide*.

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

- Configuring the Counters on page 572
- Configuring the File Information on page 572
- Configuring the Interval on page 573

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.

Copyright © 2012, Juniper Networks, Inc.

To configure which file to use, include the **file** statement at the **[edit accounting-options filter-profile *profile-name*]** hierarchy level:

```
[edit accounting-options filter-profile profile-name]
file filename;
```

You must specify a filename for the filter profile that has already been configured at the **[edit accounting-options]** hierarchy level.

NOTE: If the configured file size or transfer interval is exceeded, Junos OS closes the file and starts a new one. By default, the transfer interval value is 30 minutes. If the transfer interval is not configured, Junos OS closes the file and starts a new one when the file size exceeds its configured value or the default transfer interval value exceeds 30 minutes. To avoid transferring files every 30 minutes, specify a different value for the transfer interval.

Configuring the Interval

Each filter with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the **interval** statement at the **[edit accounting-options filter-profile *profile-name*]** hierarchy level:

```
[edit accounting-options filter-profile profile-name]
interval;
```

NOTE: The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of filters might cause serious performance degradation.

The range for the **interval** statement is 1 through 2880 minutes. The default is 30 minutes.

Related Documentation

- Accounting Options Overview on page 561
- Understanding Device Management Functions in Junos OS
- Accounting Options Configuration on page 563
- Configuring Accounting-Data Log Files on page 566
- Configuring the Interface Profile on page 569
- Configuration Statements at the [edit accounting-options] Hierarchy Level on page 5
- Example: Configuring a Filter Profile on page 574
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 574

Copyright © 2012, Juniper Networks, Inc.

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 572
- Example: Configuring Interface-Specific Firewall Counters and Filter Profiles on page 574

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.

Copyright © 2012, Juniper Networks, Inc.

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 572
- Configuring the Interface Profile on page 569

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 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. On a T4000 Type 5 FPC, the source class accounting is performed at egress. The implications of this are as follows:

- SCU accounting is *not* 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.

 Copyright © 2012, Juniper Networks, Inc.

For more information about source class usage, see the *Junos OS Policy Framework Configuration Guide*, the *Junos OS Network Interfaces Configuration Guide*, and the *Junos OS Feature Guides*.

Related Documentation

- Configuring SCU or DCU on page 577
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579
- Configuring Class Usage Profiles on page 580
- Configuring the MIB Profile on page 583
- Configuring the Routing Engine Profile on page 584

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 577
- Applying the Policy to the Forwarding Table on page 577
- Enabling Accounting on Inbound and Outbound Interfaces on page 578

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 *Junos OS Policy Framework Configuration Guide* and the *Junos OS Network Interfaces Configuration Guide*.

 Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 576
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579
- Configuring Class Usage Profiles on page 580
- Configuring the MIB Profile on page 583
- Configuring the Routing Engine Profile on page 584

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 579
- Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface on page 579
- Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface on page 580

Example: Configuring a Virtual Loopback Tunnel Interface on a Provider Edge Router Equipped with a Tunnel PIC

Define a virtual loop interface on a provider edge router with a Tunnel PIC:

```
[edit interfaces]
vt-0/3/0 {
  unit 0 {
    family inet {
      accounting {
        source-class-usage {
          input;
        }
      }
    }
  }
}
```

Example: Mapping the VRF Instance Type to the Virtual Loopback Tunnel Interface

Map the VRF instance type to the virtual loopback tunnel interface:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface at-2/1/1.0;
    interface vt-0/3/0.0;
    route-distinguisher 10.255.14.225:100;
    vrf-import import-policy-name;
    vrf-export export-policy-name;
    protocols {
```

```
            bgp {
              group to-r4 {
                local-address 10.27.253.1;
                peer-as 400;
                neighbor 10.27.253.2;
              }
            }
          }
        }
      }
```

NOTE: For SCU and DCU to work, do not include the **vrf-table-label** statement at the **[edit routing-instances *instance-name*]** hierarchy level.

Example: Sending Traffic Received from the Virtual Loopback Interface Out the Source Class Output Interface

Send traffic received from the virtual loopback tunnel interface out of the source class output interface:

```
[edit interfaces]
at-1/1/0 {
  unit 0 {
    family inet {
      accounting {
        source-class-usage {
          output;
        }
      }
    }
  }
}
```

For more information about configuring source class usage on the virtual loopback tunnel interface, see the *Junos OS Network Interfaces Configuration Guide*.

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 576
- Configuring SCU or DCU on page 577
- Configuring Class Usage Profiles on page 580
- Configuring the MIB Profile on page 583
- Configuring the Routing Engine Profile on page 584

Configuring Class Usage Profiles

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

- Configuring a Class Usage Profile on page 581
- Configuring the File Information on page 581
- Configuring the Interval on page 581

Copyright © 2012, Juniper Networks, Inc.

- Creating a Class Usage Profile to Collect Source Class Usage Statistics on page 581
- Creating a Class Usage Profile to Collect Destination Class Usage Statistics on page 582

Configuring a Class Usage Profile

You can configure the class usage profile to collect statistics for particular source and destination classes.

To configure the class usage profile to filter by source classes, include the **source-classes** statement at the **[edit accounting-options class-usage-profile *profile-name*]** hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
source-classes {
   source-class-name;
}
```

To configure the class usage profile to filter by destination classes, include the **destination-classes** statement at the **[edit accounting-options class-usage-profile *profile-name*]** hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
destination-classes {
   destination-class-name;
}
```

Configuring the File Information

Each accounting profile logs its statistics to a file in the **/var/log** directory.

To specify which file to use, include the **file** statement at the **[edit accounting-options class-usage-profile *profile-name*]** hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
file filename;
```

You must specify a filename for the source class usage profile that has already been configured at the **[edit accounting-options]** hierarchy level. You can also specify a filename for the destination class usage profile configured at the **[edit accounting-options]** hierarchy level.

Configuring the Interval

Each interface with a class usage profile enabled has statistics collected once per interval specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the **interval** statement at the **[edit accounting-options class-usage-profile *profile-name*]** hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
interval;
```

Creating a Class Usage Profile to Collect Source Class Usage Statistics

To create a class usage profile to collect source class usage statistics:

```
[edit]
accounting-options {
```

```
    class-usage-profile scu-profile1;
    file usage-stats;
    interval 15;
    source-classes {
      gold;
      silver;
      bronze;
    }
  }
```

The class usage profile, **scu-profile1**, writes data to the file **usage_stats**. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, scu_profile,epoch-timestamp,interface-name,source-class,
packet-count,byte-count
scu_profile,980313078,xe-1/0/0.0,gold,82,6888
scu_profile,980313078,xe-1/0/0.0,silver,164,13776
scu_profile,980313078,xe-1/0/0.0,bronze,0,0
scu_profile,980313678,xe-1/0/0.0,gold,82,6888
scu_profile,980313678,xe-1/0/0.0,silver,246,20664
scu_profile,980313678,xe-1/0/0.0,bronze,0,0
```

Creating a Class Usage Profile to Collect Destination Class Usage Statistics

To create a class usage profile to collect destination class usage statistics:

```
  [edit]
  accounting-options {
    class-usage-profile dcu-profile1;
    file usage-stats
    interval 15;
    destination-classes {
      gold;
      silver;
      bronze;
    }
  }
```

The class usage profile, **dcu-profile1**, writes data to the file **usage-stats**. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, dcu_profile,epoch-timestamp,interface-name,destination-class,
packet-count,byte-count
dcu_profile,980313078,xe-1/0/0.0,gold,82,6888
dcu_profile,980313078,xe-1/0/0.0,silver,164,13776
dcu_profile,980313078,xe-1/0/0.0,bronze,0,0
dcu_profile,980313678,xe-1/0/0.0,gold,82,6888
dcu_profile,980313678,xe-1/0/0.0,silver,246,20664
dcu_profile,980313678,xe-1/0/0.0,bronze,0,0
...
#FILE CLOSED 976826178 2000-12-14-20:36:18
```

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 576
- Configuring SCU or DCU on page 577
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579

Copyright © 2012, Juniper Networks, Inc.

- Configuring the Routing Engine Profile on page 584

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 583
- Configuring the Interval on page 583
- Configuring the MIB Operation on page 584
- Configuring MIB Object Names on page 584
- Example: Configuring a MIB Profile on page 584

Configuring the File Information

Each accounting profile logs its statistics to a file in the **/var/log** directory.

To configure which file to use, include the **file** statement at the **[edit accounting-options mib-profile *profile-name*]** hierarchy level:

```
[edit accounting-options mib-profile profile-name]
file filename;
```

You must specify a ***filename*** for the MIB profile that has already been configured at the **[edit accounting-options]** hierarchy level.

Configuring the Interval

A MIB profile has statistics collected once per interval time specified for the profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the **interval** statement at the **[edit accounting-options mib-profile *profile-name*]** hierarchy level:

```
[edit accounting-options mib-profile profile-name]
interval;
```

The range for the **interval** statement is 1 through 2880 minutes. The default is 30 minutes.

Configuring the MIB Operation

A MIB profile must specify the operation that is used to collect MIB statistics. To configure which operation is used to collect MIB statistics, include the **operation** statement at the **[edit accounting-options mib-profile *profile-name*]** hierarchy level:

```
[edit accounting-options mib-profile profile-name]
operation operation-name;
```

You can configure a **get**, **get-next**, or **walk** operation. The default operation is **walk**.

Configuring MIB Object Names

A MIB profile must specify the MIB objects for which statistics are to be collected. To configure the MIB objects for which statistics are collected, include the **objects-names** statement at the **[edit accounting-options mib-profile *profile-name*]** hierarchy level:

```
[edit accounting-options mib-profile profile-name]
object-names {
   mib-object-name;
}
```

You can include multiple MIB object names in the configuration.

Example: Configuring a MIB Profile

Configure a MIB profile:

```
[edit accounting-options]
mib-profile mstatistics {
   file stats;
   interval 60;
   operation walk;
   objects-names {
      ipCidrRouteStatus;
      ifOutOctets;
   }
}
```

Related Documentation

- Understanding Source Class Usage and Destination Class Usage Options on page 576
- Configuring SCU or DCU on page 577
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579
- Configuring Class Usage Profiles on page 580
- Configuring the Routing Engine Profile on page 584

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.

Copyright © 2012, Juniper Networks, Inc.

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 585
- Configuring the File Information on page 585
- Configuring the Interval on page 585
- Example: Configuring a Routing Engine Profile on page 586

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.

Copyright © 2012, Juniper Networks, Inc.

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 576
- Configuring SCU or DCU on page 577
- Configuring SCU on a Virtual Loopback Tunnel Interface on page 579
- Configuring Class Usage Profiles on page 580
- Configuring the MIB Profile on page 583

Copyright © 2012, Juniper Networks, Inc.

PART 5

Index

- Index on page 589

Copyright © 2012, Juniper Networks, Inc.

Index

Symbols

#, comments in configuration statements....xix
( ), in syntax descriptions....xix
/var/log/mib2d file....67
/var/log/snmpd file....67
< >, in syntax descriptions....xix
[ ], in configuration statements....xix
{ }, in configuration statements....xix
| (pipe), in syntax descriptions....xix

A

AAA Objects MIB....31, 201, 213
    Textual Conventions....214
Access Authentication Objects MIB....31, 201, 219
access authentication traps....217
access statement
    usage guidelines....84
access-list statement....145
accounting options
    configuration....563
    overview....561
accounting profiles
    filter....572
    interface....569
    MIB....583
    Routing Engine....584
address statement
    SNMPv3
        usage guidelines....96
address-mask statement
    usage guidelines....96
agent, SNMP....16
agent-address statement....146
Alarm MIB....31, 202, 223
Analyzer MIB....32
Antivirus Objects MIB....32
archive-sites statement
    accounting
        usage guidelines....569
ATM CoS MIB....32, 202, 227
ATM MIB....32
Auth Entry *See* Auth Entry MIB
    LSYS
        MIB....452
authentication-md5 statement
    usage guidelines....81
authentication-none statement
    usage guidelines....82
authentication-password statement
    usage guidelines....81
authentication-sha statement
    usage guidelines....81
authorization statement....146
    usage guidelines....54

B

BFD MIB....32, 202, 237
    notification variables....238
BGP4 V2 MIB....32, 202, 235
braces, in configuration statements....xix
brackets
    angle, in syntax descriptions....xix
    square, in configuration statements....xix

C

categories statement....147
    usage guidelines....60
Chassis Cluster MIB....33, 202
Chassis Definitions for Router Model MIB....32, 360
Chassis Forwarding MIB....32
Chassis MIB....33, 202
    jnxBoxAnatomy....241
    jnxBoxKernelMemoryUsedPercent....355
    jnxBoxSystemDomainType 355
    jnxMIBs....241
    jnxTraps....355
    overview....239
Class 1 MIB objects....191
Class 2 MIB objects....195
Class 3 MIB objects....196
Class 4 MIB objects....197
Class-of-Service MIB....33
class-usage-profile statement
    usage guidelines....581
client list
    adding to SNMP community....55
client-list statement....147
    usage guidelines....55
client-list-name statement....148
    usage guidelines....55

clients statement....148
usage guidelines....54
comments, in configuration statements....xix
commit-delay statement....149
usage guidelines....52
community statement
SNMP....150
usage guidelines....54
community string, SNMP....54
community-name statement
usage guidelines....106
Configuration Management MIB....33, 202, 369
contact statement....151
usage guidelines....50
conventions
text and syntax....xviii
CoS
MIB....33
CPU *See* CPU MIB
LSYS
MIB....475
curly braces, in configuration statements....xix
customer support....xx
contacting JTAC....xx

D

DCU, Destination Class Usage *See* Destination Class Usage MIB
description statement
SNMP....151
usage guidelines....51
Destination Class Usage MIB....33, 203, 375
destination-classes statement
usage guidelines....581
destination-port statement
SNMP....152
usage guidelines....60
DHCP MIB....33
DHCPv6 MIB....34
Digital Optical Monitoring MIB....34
DNS Objects MIB....34, 203, 379
Dynamic Flow Capture MIB....34
notification variables....384

E

engine-id statement
SNMPv3
usage guidelines....70, 79
enterprise-specific MIBs, listed....201
enterprise-specific traps, SNMP
version 1....126
version 2....133
Ethernet MAC MIB....34, 203, 381
Event MIB....34, 203, 383
EX Series Ethernet Switches
enterprise-specific traps....141
standard traps....142
Experimental MIB....34, 212

F

fields statement
for interface profiles
usage guidelines....570
for Routing Engine profiles
usage guidelines....585
file statement
accounting (associating with profile)
usage guidelines (filter profile)....572
usage guidelines (interface profile)....570
usage guidelines (MIB profile)....583
usage guidelines (Routing Engine profile)....585
accounting (configuring log file)
usage guidelines....566
filter profile....572
filter-duplicates statement....152
usage guidelines....51
filter-interfaces statement....153
filter-profile statement
usage guidelines....572
filtering get SNMP requests....51
Firewall MIB....35, 203, 385
Flow Collection Services MIB....35, 487
Flow Gate *See* Flow Gate MIB
LSYS
MIB....448
Flow Session *See* Flow Session MIB
Flow session
LSYS
MIB....450
font conventions....xviii

G

Get requests, SNMP....13

Copyright © 2012, Juniper Networks, Inc.

group statement
SNMPv3 (for access privileges)
usage guidelines....90
SNMPv3 (for configuring)
usage guidelines....86

H

Host Resources MIB....35, 203, 389

I

IDP attack log notification....538
IDP attack Table....537
IDP MIB....35, 537
ifChassisTable....394
informs SNMP *See* SNMP informs
Interface MIB....35, 204, 391
interface profile....569
interface statement
SNMP....154
usage guidelines....63
interface-profile statement
usage guidelines....569
interfaces limiting SNMP access....63
interval statement
accounting
usage guidelines (filter profile)....573
usage guidelines (interface profile)....570
usage guidelines (MIB profile)....583
usage guidelines (Routing Engine profile)....585
IP Forward MIB....35, 204, 397
IPsec Generic Flow Monitoring Object MIB....35, 204, 399
Textual Conventions....400
IPsec Monitoring MIB....36, 204, 415
IPsec Phase 1 IKE Tunnel Table....404
IPsec Phase 2 IKE Tunnel Table....408
IPsec Phase 2 Security Association Table....411
IPsec VPN Objects MIB....36
IPv4 MIB....36, 204, 423
IPv6 and ICMPv6 MIB....36
IPv6 SNMP community string....55

J

jnxBfdSessTable....237
jnxBgpM2PrefixCountersTable....236
jnxBoxAnatomy....241
jnxBoxKernelMemoryUsedPercent....355
jnxBoxSystemDomainType....355
jnxContainersTable
M160 router....245
M20 router....244
M40 router....244
M40e router....248
M5 router....246
T320 router....247
T4000 router....249
T640 router....247
jnxContentsTable
M20 router....251
T320 router....258
T640 router....254, 262
jnxDCUsTable....376
jnxDcuStatsTable....376
jnxEventAvTable,....383
jnxExperiment root branch....212
jnxFilledTable....271
M20 router....271
T320 router....278
T4000 router....282
T640 router....274
jnxFruTable....306
M10 router....309
M160 router....315
M20 router....311
M40 router....322
M40e router....327
T4000 router....343
T640 router....333
jnxIfTable....391
jnxIkeTunnelTable....415
jnxIPSecSaTable....421
jnxIPSecTunnelTable....418
jnxJsAuthNotifications....221
jnxJsAuthTrapVars....220
jnxJsClusterObjectsTable....543
jnxJsDnsProxyDataObjects....379
jnxJsFwAuthStats....219
jnxJsIfMonTable....519
jnxJsLoadedCaCertTable....555
jnxJsLoadedLocalCertTable....556
jnxJsNatIfSrcPoolPortTable....483
jnxJsPolicyStatsTable....509
jnxJsScreenMonTable....524
jnxJsScreenNotifications....535
jnxJsScreenTrapVars....535
jnxJsSPUMonitoringTable....542

Copyright © 2012, Juniper Networks, Inc.

jnxLEDTable....267
M20 router....269
T320 router....270
T640 router....269
jnxMacStatsTable....381
jnxMibs root branch....209
jnxOperatingTable....287
M20 router....290
T320 router....293
T4000 router....296
T640 router....290
jnxPfeNotifyGlTable....487
jnxPfeNotifyTypeTable....489
jnxPingCtlTable....492
jnxPingLastTestResultTable....502
jnxPingProbeHistoryTable....500
jnxPingResultsTable....496
jnxProducts root branch....208
jnxRedundancyTable....299
M20 router....301
T320 router....304
T4000 router....305
T640 router....302
jnxRmonAlarmGetFailure....517
jnxRmonAlarmTable....516
jnxRmonGetOk....517
jnxRpfStatsTable....513
jnxScuStatsTable....539
jnxServices root branch....208
jnxSyslogAvTable....548
jnxSyslogTable....546
jnxTraceRouteCtlTable....549
jnxTraps root branch....211
jnxUserAAAServerName....216
jnxUserAAAStatTable....216
jnxUtilCounter32Table....552
jnxUtilCounter64Table....552
jnxUtilIntegerTable....553
jnxUtilStringTable....554
jnxUtilUintTable....553
Juniper Networks MIB objects....187

L

L2ALD MIB....36
L2CP MIB....36
L2TP MIB....36
Layer 2 Control Protocol
MIB....36
LDP
MIB....37
traps....141
disabling....141
License MIB....37, 204, 433
location statement
SNMP....154
usage guidelines....50
Logical Systems MIB....37, 204, 437
logical-system statement....155
logical-system-trap-filter statement....156
LSYS
MIB.37,437,448,450,452,454,457,459,461,466,471,473,475

M

Management Information Base *See* MIBs
Management Information MIB
jnxMibs....209
jnxProducts....208
jnxServices....208
jnxTraps....211
master agent, SNMP....16
message-processing-model statement
usage guidelines....100
MIB profile....583
mib-profile statement
usage guidelines....583
MIBs
AAA Objects....31, 201, 213
Access Authentication Objects....31, 201, 219
Alarm....31, 202, 223
Analyzer....32
Antivirus Objects....32
ATM....32
ATM CoS....32, 202, 227
BFD....32, 202, 237
BGP4 V2....32, 202, 235
Chassis....33, 202, 239, 241, 355
Chassis Cluster....33, 202
Chassis Definitions for Router Model....32, 360
Chassis Forwarding....32
Class-of-Service....33
Configuration Management....33, 202, 369
Destination Class Usage....33, 203, 375
DHCP 33
DHCPv6 34
Digital Optical Monitoring....34
DNS Objects....34, 203, 379
Dynamic Flow Capture....34

Copyright © 2012, Juniper Networks, Inc.

enterprise-specific, listed....201
Ethernet MAC....34, 203, 381
Event....34, 203, 383
EX Series
Analyzer....32
PAE Extension....38
Structure of Management Information 41
Virtual Chassis....42
VLAN....42
Experimental....34, 212
Firewall....35, 203, 385
Flow Collection Services....35, 487
Host Resources....35, 203, 389
IDP....35, 537
IDP Monitoring....537
Interface....35, 204, 391
IP Forward....35, 204, 397
IPsec Generic Flow Monitoring Object 35, 204, 399
IPsec Monitoring....36, 204, 415
IPsec VPN Objects....36
IPv4....36, 204, 423
IPv6 and ICMPv6....36
L2ALD....36
L2CP 36
L2TP....36
Layer 2 Control Protocol....36
LDP....37
License....37, 433
license....204
Logical Systems....37
logical systems....204
LSYS.37,437,448,450,452,454,457,459,461,466,471,473,475
Management Information MIB
jnxMibs....209
jnxProducts....208
jnxServices....208
jnxTraps....211
MIMSTP....37
MPLS....37
MPLS LDP....37
Multicast....23, 31
NAT Objects....38, 204, 479
NAT Resources-Monitoring....38
Optical Transport Network (OTN) Interface Management 38
OSPF....20
Packet Forwarding Engine....38, 205, 487
Packet Mirror....38
PAE Extension....38
Passive Monitoring....39
Ping....39, 205, 491
interpretation of....491
use in ping test....172
view configuration example, SNMP....64
Policy Objects....39, 205, 507
Power Supply Unit....39
PPP....19, 39
PPPoE....39
Pseudowire TDM....39
Real-Time Performance Monitoring....40
Reverse-Path-Forwarding....40, 205, 513
RMON Events and Alarms 40, 205, 515
RPM....40
RSVP 40
Security Interface Extension Objects....40, 205, 519
Security Screening Objects....41, 205, 523
Services PIC....41
SNMP IDP....35
IDP Monitoring....537
SONET APS....41
SONET/SDH Interface Management....41
Source Class Usage....41, 206, 539
SPU Monitoring....41, 541
SPU monitoring....206
Structure of Management Information....41, 201
Junos OS for J Series and SRX Series devices, for....42, 201
Subscriber....42
System Log....42, 206, 545
Traceroute....42, 206
Utility....42, 206, 551
views
SNMP....66
Virtual Chassis....42
VLAN....42
VPLS 42
BGP MIB....42
Generic MIB....42
LDP MIB....42
VPN....43
VPN Certificate Objects....43, 206, 555
MIMSTP
MIB....37
minimum accounting options configuration....564

MPLS
MIB....37
standard traps....120
MPLS LDP MIB....37
Multicast MIB....23, 31

N
name statement....156
usage guidelines....52
NAT
LSYS
MIB....454, 457, 459, 461, 466, 473
NAT cone bind *See* NAT cone bind MIB
LSYS
MIB....471
NAT Objects MIB....38, 204, 479
NAT Resources-Monitoring MIB....38
NAT trap definitions....485
Network Address Translation Objects MIB *See* NAT Objects MIB
nonpersistent statement
accounting
usage guidelines....567
nonvolatile statement....157
notify statement
usage guidelines....93
notify-filter statement
for applying to target
usage guidelines....99
for configuring
usage guidelines....94
notify-view statement
usage guidelines....87
number of IKE Tunnels currently active....404
nxContainersTable
M10 router....245

O
objects-names statement
for Routing Engine profiles
usage guidelines....584
oid statement
SNMP....157
usage guidelines....66
SNMPv3
usage guidelines....94
operation statement
for MIB profiles
usage guidelines....584
Optical Transport Network (OTN) Interface Management MIB....38
OSPF MIB....20

P
Packet Forwarding Engine MIB....38, 205, 487
Packet Mirror MIB....38
PAE Extension MIB....38
parameters statement
usage guidelines....98
parentheses, in syntax descriptions....xix
Passive Monitoring MIB....39
Ping MIB....39, 205, 491
interpretation of....491
use in ping test....172
view configuration example
SNMP....64
pingProbeHistoryTable....177
Policy Objects MIB....39, 205, 507
port statement
SNMPv3
usage guidelines....96
Power Supply Unit MIB....39
PPP MIB....19, 39
PPPoE MIB....39
prefix list
adding to SNMP community....55
privacy-3des statement
usage guidelines....83
privacy-aes128 statement
usage guidelines....83
privacy-des statement
usage guidelines....83
privacy-none statement
usage guidelines....84
privacy-password statement
usage guidelines
for 3DES algorithm....83
for AES algorithm....83
for DES algorithm....83
profiles, accounting
filter....572
interface....569
MIB....583
Routing Engine....584
Pseudowire TDM MIB....39
PSU MIB....39

Copyright © 2012, Juniper Networks, Inc.

R

read-view statement
usage guidelines....88
Real-Time Performance Monitoring MIB....40
remote operations MIBs....171
request snmp spoof-trap command....63
retry-count statement
usage guidelines....104
Reverse-Path-Forwarding MIB....40, 205, 513
RMON Events and Alarms MIB....40, 205, 515
Routing Engine profile....584
routing instances
access lists
configuring....186
SNMP
enabling access....183
identifying....182
specifying....184
routing-engine-profile statement
usage guidelines....584
routing-instance statement
SNMP....158
SNMPv3
usage guidelines....96
routing-instance-access....159
RPM MIB....40
RSVP MIB....40

S

SCU, Source Class Usage *See* Source Class Usage MIB
Security Interface Extension Objects MIB....40, 205, 519
Security Policy Table....508
Security Profile *See* Security Profile MIB
MIB....437
Security Screening Objects MIB....41, 205, 523
security-level statement
for access privileges
usage guidelines....86
for SNMP notifications
usage guidelines....100
security-model statement
for access privileges
usage guidelines....86
for groups
usage guidelines....90
for SNMP notifications
usage guidelines....100
security-name statement
for security group
usage guidelines....90
for SNMP notifications
usage guidelines....101
security-to-group statement
usage guidelines....84
Services PIC MIB....41
Set requests, SNMP....13
size statement
accounting
usage guidelines....567
SNMP
adding client lists and prefix lists....55
agent....13, 16
architecture....13
commit delay timer....52
community string....54
configuration
version 3....76
versions 1 and 2....48
enterprise-specific traps *See* SNMP traps
filtering duplicate requests....51
limiting interface access....63
logging, enabling....172
manager....13
master agent....16
MIB views....66
remote operations....169
spoofing traps....63
standard traps *See* SNMP traps
standards documents....17
subagent....16
system contact....50
system description....51
system location....50, 154
system name....52
tracing operations....67
trap groups....60
trap notification for remote operations....170
trap options....57
views, setting....170
SNMP IDP MIB....537
SNMP informs....101
snmp statement....159
usage guidelines
SNMPv1 and SNMPv2....48
SNMPv3....76

Copyright © 2012, Juniper Networks, Inc.

SNMP traps....14
enterprise-specific
version 1....126
version 2....133
EX Series Ethernet Switches....141, 142
MAC limit....142
MX960 3D Universal Edge Router....142
source address configuration....58
spoofing....63
standard
version 1....113
version 2....117
system logging severity levels....15
unsupported....121
SNMPv2
MPLS traps....120
Passive Monitoring Traps MIB....60
SNMPv3
authentication, configuring....81
informs, configuring....101
local engine ID, configuring....70, 79
minimum configuration....78
SONET APS MIB....41
SONET Automatic Protection Switching MIB....41
SONET/SDH Interface Management MIB....41
Source Class Usage MIB....41, 206, 539
Source NAT Table....480
source-address statement....160
usage guidelines....58
source-classes statement
usage guidelines....581
SPU Monitoring MIB....41, 541
SPU monitoring MIB....206
SRX 3600 Services Gateway
MIB objects....363
SRX1400 Services Gateway
MIB objects....361
SRX3400 Services Gateway
MIB objects....362
standard traps, SNMP
version 1....113
version 2....117
standards documents
SNMP and MIBs....18
start-time statement
usage guidelines....568
Structure of Management Information MIB....41, 201
for EX Series....41
Junos OS for J Series and SRX Series devices, for....42, 201
subagent, SNMP....16
Subscriber MIB....42
support, technical *See* technical support
syntax conventions....xviii
sysContact object, MIB II....50
sysDescription object, MIB II....51
sysLocation object, MIB II....50
sysName object, MIB II....52
system contact, SNMP....50
system description, SNMP....51
system location, SNMP....50, 154
System Log MIB....42, 206, 545
system logging severity levels, SNMP traps....15
system name, SNMP....52

T

tag statement
SNMPv3
usage guidelines....107
usage guidelines....93
tag-list statement
usage guidelines....96
target-address statement
usage guidelines....95
target-parameters statement
usage guidelines....98
targets statement....161
usage guidelines....60
technical support
contacting JTAC....xx
timeout statement
usage guidelines....104
traceoptions statement....162
SNMP
usage guidelines....67
Traceroute MIB....42, 206
tracing operations
SNMP....67
transfer-interval statement
accounting
usage guidelines....568
trap groups, SNMP....60
trap notification for SNMP remote operations....170
trap-group statement....164
usage guidelines....60

Copyright © 2012, Juniper Networks, Inc.

trap-options statement....165
usage guidelines....57
traps....133
definition....14
LDP....141
SNMP version 1 traps
enterprise-specific....126
standard....113
SNMP version 2 traps
enterprise-specific....133
standard....117
spoofing SNMP traps....63
unsupported....121
See also SNMP traps
type statement
usage guidelines....93

U

unsupported standard SNMP traps....121
Utility MIB....42, 206, 551

V

v3 statement
usage guidelines....76
vacm statement
usage guidelines....84
var/log/mib2d file....67
var/log/snmpd file....67
variable-length string indexes....171
version statement
SNMP....166
usage guidelines....60
view statement
SNMP (associating with community)....167
usage guidelines....54
SNMP (configuring MIB view)....168
usage guidelines....66
views, MIB
SNMP....66, 170
Virtual Chassis MIB....42
VLAN MIB....42
VPLS MIBs....42
VPN Certificate Objects MIB....43, 206, 555
VPN MIB....43

W

write-view statement
usage guidelines....88

Copyright © 2012, Juniper Networks, Inc.