



Junos[®] OS

SNMP MIBs and Traps Reference

Release
13.2



Published: 2013-07-23

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

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

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

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

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

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

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

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

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

Junos® OS SNMP MIBs and Traps Reference

13.2

Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xxxv
	Documentation and Release Notes	xxxv
	Supported Platforms	xxxv
	Using the Examples in This Manual	xxxv
	Merging a Full Example	xxxvi
	Merging a Snippet	xxxvi
	Documentation Conventions	xxxvii
	Documentation Feedback	xxxix
	Requesting Technical Support	xxxix
	Self-Help Online Tools and Resources	xxxix
	Opening a Case with JTAC	xl
Part 1	Overview	
Chapter 1	Getting Started	3
	Understanding the SNMP Implementation in Junos OS	3
	SNMP Architecture	3
	SNMP MIBs	4
	SNMP Traps and Informs	4
	Junos OS SNMP Agent Features	6
	Loading MIB Files to a Network Management System	7
Chapter 2	SNMP MIBs	9
	Standard SNMP MIBs Supported by Junos OS	9
	Juniper Networks Enterprise-Specific MIBs	25
	Juniper Networks Enterprise-Specific MIBs and Supported Devices	36
	SNMP MIB Objects Supported by Junos OS for the Set Operation	46
	LLDP Standard MIB Objects Supported on EX Series Devices	53
Chapter 3	SNMP Notifications	57
	Standard SNMP Traps Supported on Devices Running Junos OS	57
	Juniper Networks Enterprise-Specific SNMP Traps	57
	Managing Traps and Informs	58
	Generating Traps Based on SysLog Events	58
	Filtering Traps Based on the Trap Category	59
	Filtering Traps Based on the Object Identifier	59

Part 2	Enterprise-Specific MIBs	
Chapter 4	AAA Objects MIB	67
	Interpreting the Enterprise-Specific AAA Objects MIB	67
	AAA Objects MIB	67
	AAA Objects MIB Textual Conventions	68
	jnxUserAAASatTable	69
	jnxUserAAAServerName	70
	Access Authentication-Related Traps	70
	jnxUserAAAAccessPool	71
Chapter 5	Access Authentication Objects MIB	73
	Interpreting the Enterprise-Specific Access Authentication Objects MIB	73
	Access Authentication Objects MIB	73
	jnxJsFwAuthStats	73
	jnxJsAuthTrapVars	74
	jnxJsAuthNotifications	74
Chapter 6	Alarm MIB	77
	Interpreting the Enterprise-Specific Alarm MIB	77
	Alarm MIB	77
	jnxAlarmRelayMode	77
	jnxYellowAlarms	78
	jnxRedAlarms	78
Chapter 7	Analyzer MIB	81
	Interpreting the Enterprise-Specific Analyzer MIB	81
	Analyzer MIB	81
	jnxAnalyzerTable	81
	jnxAnalyzerInputTable	82
	jnxAnalyzerOutputTable	83
Chapter 8	Antivirus Objects MIB	85
	Interpreting the Enterprise-Specific Antivirus Objects MIB	85
	Antivirus Objects MIB	85
	Scan Engine Objects Table	86
	Scan Statistics Objects Table	86
	Antivirus Traps Objects Table	87
Chapter 9	ATM Class-of-Service MIB	89
	Interpreting the Enterprise-Specific ATM Class-of-Service MIB	89
	ATM Class-of-Service MIB	89
	jnxCosAtmVcTable	90
	jnxCosAtmVcScTable	90
	jnxCosAtmVcQstatsTable	92
	jnxCosAtmTrunkTable	93
Chapter 10	ATM MIB	95
	Interpreting the Enterprise-Specific ATM MIB	95
	ATM MIB	95
	jnxAtmIfTable	95
	jnxAtmVCTable	98

	jnxAtmVpTable	100
	jnxAtmTrunkTable	102
Chapter 11	BGP4 V2 MIB	105
	Interpreting the Enterprise-Specific BGP4 V2 MIB	105
	BGP4 V2 MIB	105
	jnxBgpM2PrefixCountersTable	105
	Enterprise-Specific BGP Traps	106
Chapter 12	Bidirectional Forwarding Detection MIB	107
	Interpreting the Enterprise-Specific Bidirectional Forwarding Detection MIB	107
	Bidirectional Forwarding Detection MIB	107
	jnxBfdSessTable	107
	Notifications for the BFD MIB	108
Chapter 13	Chassis Cluster MIB	109
	Interpreting the Enterprise-Specific Chassis Cluster MIB	109
	Chassis Cluster MIB	109
	jnxJsChassisClusterSwitchover	109
	jnxJsChClusterIntfTrap	110
	Chassis Cluster Notifications	111
Chapter 14	Chassis Forwarding MIB	113
	Interpreting the Enterprise-Specific Chassis Forwarding MIB	113
	Chassis Forwarding MIB	113
	jnxFwddProcess	113
Chapter 15	Chassis MIB	115
	Interpreting the Enterprise-Specific Chassis MIBs	115
	Chassis MIBs	116
	Chassis MIB Textual Conventions	117
	jnxBoxAnatomy	118
	Top-Level Objects	118
	jnxContainersTable	119
	jnxContentsLastChange	134
	jnxContentsTable	134
	jnxLEDLastChange	174
	jnxLEDTable	174
	jnxFilledLastChange	184
	jnxFilledTable	184
	jnxOperatingTable	215
	jnxRedundancyTable	250
	jnxFruTable	258
	jnxFruMib	345
	jnxBoxKernelMemoryUsedPercent	346
	jnxBoxSystemDomainType	346
	Chassis Traps	346
	SNMPv1 Trap Format	349
	SNMPv2 Trap Format	350
	Chassis Definitions for Router Model MIB	352
	MIB Objects for the M120 Router	354

	MIB Objects for the MX2020 3D Universal Edge Router	356
	MIB Objects for the MX2010 3D Universal Edge Router	357
	MIB Objects for the MX960 3D Universal Edge Router	358
	MIB Objects for the MX480 3D Universal Edge Router	358
	MIB Objects for the MX240 3D Universal Edge Router	358
	MIB Objects for the MX80 3D Universal Edge Router	359
	MIB Objects for the MX40 3D Universal Edge Router	359
	MIB Objects for the MX10 3D Universal Edge Router	360
	MIB Objects for the MX5 3D Universal Edge Router	360
	MIB Objects for the PTX5000 Packet Transport Router	361
	MIB Objects for the QFX Series	362
	QFX3500 and QFX3600 Switches	363
	QFabric Systems	363
	QFabric System QFX3100 Director Device	363
	QFabric System QFX3008-I Interconnect Device	363
	QFabric System QFX3600-I Interconnect Device	364
	QFabric System Node Devices	364
	MIB Objects for the SRX1400 Services Gateway	364
	MIB Objects for the SRX3400 Services Gateway	365
	MIB Objects for the SRX3600 Services Gateway	365
	MIB Objects for the SRX5600 Services Gateway	366
	MIB Objects for the SRX5800 Services Gateway	366
	MIB Objects for the SRX100 Services Gateway	367
	MIB Objects for the SRX210 Services Gateway	367
	MIB Objects for the SRX220 Services Gateway	368
	MIB Objects for the SRX240 Services Gateway	368
	MIB Objects for the SRX550 Services Gateway	368
	MIB Objects for the SRX650 Services Gateway	369
Chapter 16	Class-of-Service MIB	371
	Interpreting the Enterprise-Specific Class-of-Service MIB	371
	Class-of-Service MIB	371
	jnxCosInvQstatTable	371
Chapter 17	Configuration Management MIB	375
	Interpreting the Enterprise-Specific Configuration Management MIB	375
	Configuration Management MIB	375
	Configuration Management MIB Textual Conventions	375
	Configuration Change Management Objects and	
	jnxCmCfgChgEventTable	376
	jnxCmCfgChgEventTable	377
	Rescue Configuration Change Management Objects	378
	Configuration Management Notifications	378
Chapter 18	Destination Class Usage MIB	381
	Interpreting the Enterprise-Specific Destination Class Usage MIB	381
	Destination Class Usage MIB	381
	jnxDCUsTable	381
	jnxDcuStatsTable	382

Chapter 19	DHCP MIB	385
	Interpreting the Enterprise-Specific DHCP MIB	385
	DHCP MIB	385
	DHCP Local Server	385
	DHCP Relay	391
Chapter 20	DHCPv6 MIB	397
	Interpreting the Enterprise-Specific DHCPv6 MIB	397
	DHCPv6 MIB	397
	DHCPv6 Local Server	397
Chapter 21	Digital Optical Monitoring MIB	405
	Interpreting the Enterprise-Specific Digital Optical Monitoring MIB	405
	Digital Optical Monitoring MIB	405
	DOM MIB Textual Conventions	405
	jnxDomCurrentTable	406
	jnxDomNotifications	408
Chapter 22	DNS Objects MIB	411
	Interpreting the Enterprise-Specific DNS Objects MIB	411
	DNS Objects MIB	411
	jnxJsDnsProxyDataObjects	411
Chapter 23	Dynamic Flow Capture MIB	413
	Interpreting the Enterprise-Specific Dynamic Flow Capture MIB	413
	Dynamic Flow Capture MIB	413
	jnxDfcCSTable	414
	jnxDfcCDTable	417
	DFC Notification Variables	418
	DFC Notification Definitions	419
Chapter 24	Ethernet MAC MIB	421
	Interpreting the Enterprise-Specific Ethernet MAC MIB	421
	Ethernet MAC MIB	421
	jnxMacStatsTable	421
Chapter 25	Event MIB	423
	Interpreting the Enterprise-Specific Event MIB	423
	Event MIB	423
	jnxEventAvTable	423
	Notifications for the Event MIB	424
Chapter 26	Experiment MIB	425
	jnxExperiment	425
Chapter 27	EX Series MAC Notification MIB	427
	Interpreting the Enterprise-Specific EX Series MAC Notification MIB	427
	EX Series MAC Notification MIB	427
	MAC Notification Global Group	427
	MAC Notification MIB Global Objects	428
	MAC History Index	429
	MAC History Changes Message	429

Chapter 28	EX Series SMI MIB	431
	Interpreting the Enterprise-Specific EX Series SMI MIB	431
	EX Series SMI MIB	431
Chapter 29	EX Series Virtual Chassis MIB	433
	Interpreting the Enterprise-Specific Virtual Chassis MIB	433
	Virtual Chassis MIBs	433
	jnxVirtualChassisMemberTable	433
Chapter 30	Firewall MIB	435
	Interpreting the Enterprise-Specific Firewall MIB	435
	Firewall MIB	435
	jnxFirewallsTable	436
	jnxFirewallCounterTable	436
Chapter 31	Flow Collection Services MIB	439
	Interpreting the Enterprise-Specific Flow Collection Services MIB	439
	Flow Collection Services MIB	439
	jnxCollGlobalStats	439
	jnxCollPicIfTable	440
	jnxCollFileTable	442
Chapter 32	Host Resources MIB	445
	Interpreting the Enterprise-Specific Host Resources MIB	445
	Host Resources MIB	445
	jnxHrStorageTable	445
Chapter 33	Interface MIB	447
	Interpreting the Enterprise-Specific Interface MIB	447
	Interface MIB	447
	jnxIfTable	447
	ifChassisTable	450
Chapter 34	IP Forward MIB	453
	Interpreting the Enterprise-Specific IP Forward MIB	453
	IP Forwarding MIB	453
	jnxIpCidrRouteTable	454
	jnxInetCidrRouteTable	454
Chapter 35	IPsec Generic Flow Monitoring Object MIB	457
	Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object	
	MIB	457
	IPsec Generic Flow Monitoring Object MIB	457
	Branch Tree Objects	458
	IPsec Generic Flow Monitoring Object MIB Textual Conventions	458
	Number of IKE Tunnels Currently Active	462
	IPsec Phase 1 IKE Tunnel Table	462
	IPsec Phase 2 IKE Tunnel Table	465
	IPsec Phase 2 Security Association Table	469

Chapter 36	IPsec Monitoring MIB	473
	Interpreting the Enterprise-Specific IPsec Monitoring MIB	473
	IPsec Monitoring MIB	473
	jnxIkeTunnelTable	473
	jnxIPSecTunnelTable	476
	jnxIPSecSaTable	478
Chapter 37	IPsec VPN Objects MIB	481
	Interpreting the Enterprise-Specific IPsec VPN Objects MIB	481
	IPsec VPN Objects MIB	481
	IPsec VPN Objects MIB Textual Conventions	482
	jnxJslpSecTunnelTable	482
Chapter 38	IPv4 MIB	485
	Interpreting the Enterprise-Specific IPv4 MIB	485
	IPv4 MIB	485
	jnxIpv4AddrTable	485
	Standard IPv4/IPv6 MIBs	486
Chapter 39	IPv6 MIB	507
	Interpreting the Enterprise-Specific IPv6 MIB	507
	IPv6 MIB	507
	jnxIpv6GlobalStats	507
	jnxIcmpv6GlobalStats	510
	jnxIpv6IfStatsTable	514
	Standard IPv4/IPv6 MIBs	514
Chapter 40	L2ALD MIB	537
	Interpreting the Enterprise-Specific L2ALD MIB	537
	L2ALD MIB	537
	jnxL2aldInterfaceTable	537
	MAC Address Limit Traps	538
Chapter 41	L2CP MIB	541
	Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB	541
	L2CP MIB	541
	L2CP MIB Objects Supported by the Junos OS	541
Chapter 42	L2TP MIB	543
	Interpreting the Enterprise-Specific L2TP MIB	543
	L2TP MIB	543
	L2TP Scalar Status and Statistics Group	543
	jnxL2tpTunnelGroupStatsTable	544
	jnxL2tpTunnelStatsTable	545
	jnxL2tpSessionStatsTable	547
	jnxL2tpMlpppBundleStatsTable	554

Chapter 43	LDP MIB	557
	Interpreting the Enterprise-Specific LDP MIB	557
	LDP MIB	557
	Juniper Networks Enterprise-Specific LDP Traps	558
	LDP Notification Objects and Notification Types	558
	Disabling LDP Traps	560
	jnxLdpStatsTable	560
Chapter 44	License MIB	563
	Interpreting the Enterprise-Specific License MIB	563
	License MIB	563
	jnxLicenseInstallTable	564
	Supported Features and Configuration Parameters for Licensing	564
	jnxLicenseFeatureListTable	564
	License Configuration Parameters	565
	Juniper Networks Enterprise-Specific License MIB Notifications	565
Chapter 45	Logical Systems MIBs	567
	Interpreting the Enterprise-Specific Logical Systems (LSYS) MIBs	567
	LSYS Security Profile	567
	jnxLsysSecurityProfile	567
	LSYS Zone MIB	568
	jnxLsysSpZoneMIB	568
	jnxLsysSpZoneObjects	569
	jnxLsysSpZoneSummary	569
	LSYS Scheduler MIB	570
	jnxLsysSpSchedulerMIB	570
	jnxLsysSpSchedulerObjects	571
	jnxLsysSpSchedulerSummary	571
	LSYS Policy MIB	572
	jnxLsysSpPolicyMIB	572
	jnxLsysSpPolicyObjects	573
	jnxLsysSpPolicySummary	573
	LSYS Policy with Count MIB	574
	jnxLsysSpPolicywcntMIB	574
	jnxLsysSpPolicywcntObjects	575
	jnxLsysSpPolicywcntSummary	575
	LSYS Flow Gate MIB	576
	jnxLsysSpFlowgateMIB	577
	jnxLsysSpFlowgateObjects	577
	jnxLsysSpFlowgateSummary	578
	LSYS Flow Session MIB	578
	jnxLsysSpFlowsessMIB	578
	jnxLsysSpFlowsessObjects	579
	jnxLsysSpFlowsessSummary	579
	LSYS AUTH Entry MIB	580
	jnxLsysSpAuthentryMIB	580
	jnxLsysSpAuthentryObjects	581
	jnxLsysSpAuthentrySummary	581

LSYS NAT Source Pool MIB	582
jnxLsysSpNATsrcpoolMIB	582
jnxLsysSpNATsrcpoolObjects	583
jnxLsysSpNATsrcpoolSummary	583
LSYS NAT Destination Pool MIB	584
jnxLsysSpNATdstpoolMIB	584
jnxLsysSpNATdstpoolObjects	585
jnxLsysSpNATdstpoolSummary	585
LSYS NAT Source PAT Address MIB	586
jnxLsysSpNATsrcpatadMIB	586
jnxLsysSpNATsrcpatadObjects	587
jnxLsysSpNATsrcpatadSummary	587
LSYS NAT Source No PAT Address MIB	588
jnxLsysSpNATsrcnopatadMIB	588
jnxLsysSpNATsrcnopatadObjects	589
jnxLsysSpNATsrcnopatadSummary	589
LSYS NAT Source Rule MIB	590
jnxLsysSpNATsrcruleMIB	590
jnxLsysSpNATsrcruleObjects	591
jnxLsysSpNATsrcruleSummary	591
LSYS NAT Destination Rule MIB	592
jnxLsysSpNATdstruleMIB	593
jnxLsysSpNATdstruleObjects	593
jnxLsysSpNATdstruleSummary	594
LSYS NAT Static Rule MIB	595
jnxLsysSpNATstaticruleMIB	595
jnxLsysSpNATstaticruleObjects	595
jnxLsysSpNATstaticruleSummary	596
LSYS NAT Cone Bind MIB	597
jnxLsysSpNATconebindMIB	597
jnxLsysSpNATconebindObjects	597
jnxLsysSpNATconebindSummary	598
LSYS NAT Port Overloading IP Number MIB	599
jnxLsysSpNATpoipnumMIB	599
jnxLsysSpNATpoipnumObjects	599
jnxLsysSpNATpoipnumSummary	600
LSYS CPU MIB	601
jnxLsysSpCPUMIB	601
jnxLsysSpCPUObjects	601
jnxLsysSpCPSummary	602
jnxLsysSpSPUSummary	603
Chapter 46	
MIMSTP MIB	605
Interpreting the Enterprise-Specific MIMSTP MIB	605
MIMSTP MIB	605
jnxMIDot1sJuniperMstTable	606
jnxMIMstMstiBridgeTable	612
jnxMIMstVlanInstanceMappingTable	614
jnxMIMstCistPortTable	615

	jnxMIMstMstiPortTable	618
	Juniper Networks Enterprise-Specific MIMSTP Traps	620
Chapter 47	MPLS LDP MIB	623
	Interpreting the Enterprise-Specific MPLS LDP MIB	623
	MPLS LDP MIB	623
Chapter 48	MPLS MIB	625
	Interpreting the Enterprise-Specific MPLS MIB	625
	MPLS MIB	625
	mplsInfo	625
	mplsTEInfo	626
	mplsAdminGroup	626
	mplsLspInfoList	627
	Juniper Networks Enterprise-Specific MPLS Traps	630
Chapter 49	NAT Objects MIB	633
	Interpreting the Enterprise-Specific Network Address Translation Objects	
	MIB	633
	NAT Objects MIB	633
	Source NAT Table	634
	New Source NAT Table	635
	jnxJsNatIfSrcPoolPortTable	636
	NAT Rule Hit Table	637
	NAT Pool Hit Table	637
	Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services	
	Gateways	638
Chapter 50	Network Address Translation Resources—Monitoring MIB	639
	Interpreting the Enterprise-Specific Network Address Translation	
	Resources—Monitoring MIB	639
	Network Address Translation Resources—Monitoring MIB	639
	Source NAT Table	640
	NAT Pool Hit Table	641
	NAT Rule Hit Table	641
	Juniper Networks Enterprise-Specific NAT Traps on MX Series 3D Universal	
	Edge Routers	642
Chapter 51	OTN Interface Management MIB	643
	Interpreting the Enterprise-Specific OTN Interface Management MIB	643
	OTN Interface Management MIB	643
	OTN Interface Management MIB Textual Conventions	644
	jnxOtnAlarmTable	645
	jnxOtnCurrentOdu15minTable	645
Chapter 52	Packet Forwarding Engine MIB	647
	Interpreting the Enterprise-Specific Packet Forwarding Engine MIB	647
	Packet Forwarding Engine MIB	647
	jnxPfeNotifyGfTable	647
	jnxPfeNotifyTypeTable	649

Chapter 53	Packet Mirror MIB	651
	Interpreting the Enterprise-Specific Packet Mirror MIB	651
	Packet Mirror MIB Overview	651
	jnxJsPacketMirrorTrapVars	651
	Packet Mirroring-Related Traps	653
Chapter 54	PAE Extension MIB	655
	Interpreting the Enterprise-Specific PAE Extension MIB	655
	PAE Extension MIB	655
	jnxAuthProfileName	655
	jnxPaeAuthConfigTable	656
	jnxStaticMacAuthBypassTable	656
	jnxStaticMacAuthBypassIfTable	657
Chapter 55	Passive Monitoring MIB	659
	Interpreting the Enterprise-Specific Passive Monitoring MIB	659
	Passive Monitoring MIB	659
	jnxPMonFlowTable	660
Chapter 56	PING MIB	663
	Interpreting the Enterprise-Specific Ping MIB	663
	PING MIB	663
	jnxPingCtlTable	664
	jnxPingResultsTable	668
	jnxPingProbeHistoryTable	671
	jnxPingLastTestResultTable	673
Chapter 57	Policy Objects MIB	679
	Interpreting the Enterprise-Specific Policy Objects MIB	679
	Policy Objects MIB	679
	Security Policy Table	680
	jnxJsPolicyStatsTable	681
Chapter 58	Power Supply Unit MIB	685
	Interpreting the Enterprise-Specific Power Supply Unit MIB	685
	Power Supply Unit MIB	685
	Power Supply Unit Environment Group	685
	Power Supply Unit Device Power Group	686
	Power Supply Unit Outlets Group	686
	Power Source Equipment Objects Group	687
	Power Supply Unit FPC Power Allocated Information	688
Chapter 59	PPP MIB	689
	Interpreting the Enterprise-Specific PPP MIB	689
	PPP MIB	689
	PPP MIB Textual Conventions	690
	Managed Objects for PPP Link Control Protocol Layer	691
	jnxPppLinkStatusTable	691
	jnxPppLinkConfigTable	694
	jnxPppNextIfIndex	696

	Managed Objects for PPP IP Network Control Protocol Layer	696
	jnxPppIpTable	696
	jnxPppIpConfigTable	697
	Managed Objects for PPP OSI Network Control Protocol Layer	698
	jnxPppOsiTable	699
	jnxPppOsiConfigTable	700
	Managed Objects for PPP Session Attributes	701
	Managed Objects for Multilink PPP Layer	703
	jnxPppMlPppBundleTable	703
	jnxPppMlPppNextLinkIfIndex	703
	jnxPppMlPppLinkConfigTable	704
	jnxPppMlPppNextNetworkIfIndex	706
	jnxPppMlPppNetworkConfigTable	707
	jnxPppMlPppLinkBindTable	707
	Managed Objects for PPP Interface Summary Counts	708
	Managed Objects for PPP IPv6 Network Control Protocol Layer	713
	jnxPppIpv6Table	714
	jnxPppIpv6ConfigTable	715
	Managed Objects for PPP Global Attributes	715
Chapter 60	PPPoE MIB	717
	Interpreting the Enterprise-Specific PPPoE MIB	717
	PPPoE MIB	717
	PPPoE MIB Textual Conventions	718
	Managed Objects for PPPoE Underlying Interfaces Layer Functions	718
	jnxPPPoElfTable	718
	jnxPPPoElfStatsTable	720
	jnxPPPoElfLockoutTable	721
	Managed Objects for PPPoE Interfaces Layer Functions	722
	jnxPPPoESubIfNextIndex	723
	jnxPPPoESubIfTable	723
	jnxPPPoESubIfQueueStatsTable	724
	Managed Objects for PPPoE Summary Counts	724
	Managed Objects for PPPoE ServiceName Tables	725
	jnxPPPoEServiceNameTableTable	726
	jnxPPPoEServiceNameTable	726
	jnxPPPoEServiceNameAciAriTable	727
Chapter 61	Pseudowire ATM MIB	729
	Interpreting the Enterprise-Specific Pseudowire ATM MIB	729
	Pseudowire ATM MIB	729
	jnxpwAtmCfgTable	730
	jnxpwAtmOutboundNtoITable	731
	jnxpwAtmInboundNtoITable	733
	jnxpwAtmPerfCurrentTable	735
	jnxpwAtmPerfIntervalTable	736
	jnxpwAtmPerf1DayIntervalTable	738

Chapter 62	Pseudowire TDM MIB	741
	Interpreting the Enterprise-Specific Pseudowire TDM MIB	741
	Pseudowire TDM MIB	741
	Pseudowire TDM MIB Textual Conventions	742
	jnxdpwTDMTable	742
	jnxdpwTDMCfgTable	744
	jnxdpwTDMPerfCurrentTable	746
	jnxdpwTDMPerfIntervalTable	747
	jnxdpwTDMPerfDayIntervalTable	748
Chapter 63	Real-Time Performance Monitoring MIB	751
	Interpreting the Enterprise-Specific Real-Time Performance Monitoring MIB	751
	Real-Time Performance Monitoring MIB	751
	jnxdRpmResultsSampleTable	752
	JnxdRpmMeasurementType	752
	JnxdRpmTimestampType	754
	jnxdRpmResultsSummaryTable	754
	jnxdRpmResultsCalculatedTable	755
	jnxdRpmHistorySampleTable	756
	jnxdRpmHistorySummaryTable	756
	jnxdRpmHistoryCalculatedTable	757
Chapter 64	Reverse Path Forwarding MIB	759
	Interpreting the Enterprise-Specific Reverse Path Forwarding MIB	759
	Reverse Path Forwarding MIB	759
	jnxdRpfStatsTable	759
Chapter 65	RMON Events and Alarms MIB	761
	Interpreting the Enterprise-Specific RMON Events and Alarms MIB	761
	RMON Events and Alarms MIB	761
	jnxdRmonAlarmTable	761
	RMON Event and Alarm Traps	763
Chapter 66	RSVP MIB	765
	Interpreting the Enterprise-Specific RSVP MIB	765
	RSVP MIB	765
	jnxdRsvpSessionTable	765
Chapter 67	Security Interface Extension Objects MIB	769
	Interpreting the Enterprise-Specific Security Interface Extension Objects MIB	769
	Security Interface Extension Objects MIB	769
	jnxdJsIfMonTable	769
Chapter 68	Security Screening Objects MIB	773
	Interpreting the Enterprise-Specific Security Screening Objects MIB	773
	Security Screening Objects MIB	773
	jnxdJsScreenMonTable	774
	jnxdJsScreenSweepTable	785
	jnxdJsScreenNotifications	786
	jnxdJsScreenTrapVars	786

Chapter 69	Services PIC MIB	789
	Interpreting the Enterprise-Specific Services PIC MIB	789
	Services PIC MIB	789
	jnxSpSvcSetTable	790
	jnxSpSvcSetSvcTypeTable	791
	jnxSpSvcSetIfTable	792
	Service Traps	793
	Redundant Interfaces	793
Chapter 70	SNMP IDP MIB	795
	Interpreting the Enterprise-Specific SNMP IDP MIB	795
	SNMP IDP MIB	795
	jnxJsldpObjects	795
	jnxJsldpAttackTable	796
	IDP Signature Update and Attack Traps	797
Chapter 71	SONET/SDH Interface Management MIB	799
	Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB . .	799
	SONET/SDH Interface Management MIB	799
	jnxSonetAlarmsTable	799
Chapter 72	SONET APS MIB	803
	Interpreting the Enterprise-Specific SONET APS MIB	803
	SONET APS MIB	803
	apsConfigTable	804
	apsStatusTable	805
	apsChanConfigTable	808
	apsChanStatusTable	809
Chapter 73	Source Class Usage MIB	813
	Interpreting the Enterprise-Specific Source Class Usage MIB	813
	Source Class Usage MIB	813
	jnxScuStatsTable	813
Chapter 74	SPU Monitoring Objects MIB	815
	Interpreting the Enterprise-Specific SPU Monitoring Objects MIB	815
	SPU Monitoring Objects MIB	815
	SPU Monitoring Objects Table	816
	Cluster Objects Table	817
Chapter 75	Structure of Management Information MIB	819
	Interpreting the Enterprise-Specific Structure of Management Information	
	MIB	819
	Structure of Management Information MIB	819
	jnxProducts	819
	jnxServices	820
	jnxMibs	821
	jnxTraps	822
	jnxExperiment	823

Chapter 76	Subscriber MIB	825
	Interpreting the Enterprise-Specific Subscriber MIB	825
	Subscriber MIB	825
	Subscriber MIB Textual Conventions	826
	Managed Objects for General Subscriber Functions and	
	jnxSubscriberTable	826
	jnxSubscriberTable	827
	Managed Objects for Logical System Subscriber Information and	
	jnxSubscriberLogicalSystemTable	828
	jnxSubscriberLogicalSystemTable	828
	Managed Objects for Routing Instance Subscriber Information and	
	jnxSubscriberRoutingInstanceTable	829
	jnxSubscriberRoutingInstanceTable	829
Chapter 77	System Log MIB	831
	Interpreting the Enterprise-Specific System Log MIB	831
	System Log MIB	831
	jnxSyslogTable	831
	jnxSyslogAvTable	834
Chapter 78	Traceroute MIB	835
	Interpreting the Enterprise-Specific Traceroute MIB	835
	Traceroute MIB	835
	jnxTraceRouteCtlTable	835
Chapter 79	Utility MIB	837
	Interpreting the Enterprise-Specific Utility MIB	837
	Utility MIB	837
	jnxUtilCounter32Table	838
	jnxUtilCounter64Table	838
	jnxUtilIntegerTable	838
	jnxUtilUintTable	839
	jnxUtilStringTable	839
Chapter 80	VLAN MIB	841
	Interpreting the Enterprise-Specific VLAN MIB	841
	VLAN MIB	841
	VLAN Configuration Table	842
	jnxExVlanTable	842
	VLAN Interface Table	843
	jnxExVlanInterfaceTable	844
	Port Group Table	844
	jnxExVlanPortGroupTable	845
	MAC List Table	846
Chapter 81	VPLS MIB	849
	Interpreting the Enterprise-Specific VPLS MIBs	849
	Juniper Networks Enterprise-Specific Implementation of VPLS MIBs	849
	Juniper Networks Enterprise-Specific VPLS Generic MIB	850
	jnxVplsConfigTable	850
	jnxVplsStatusTable	851

	jnxVplsPwBindTable	852
	jnxVplsBgpADConfigTable	853
	jnxVplsBgpRteTargetTable	854
	Juniper Networks Enterprise-Specific VPLS LDP MIB	854
	jnxVplsLdpConfigTable	855
	jnxVplsLdpPwBindTable	855
	Juniper Networks Enterprise-Specific VPLS BGP MIB	855
	jnxVplsBgpConfigTable	855
	jnxVplsBgpVETable	856
	jnxVplsBgpPwBindTable	856
Chapter 82	VPN Certificate Objects MIB	859
	Interpreting the Enterprise-Specific VPN Certificate Objects MIB	859
	VPN Certificate Objects MIB	859
	jnxJsLoadedCaCertTable	859
	jnxJsLoadedLocalCertTable	860
Chapter 83	VPN MIB	863
	Interpreting the Enterprise-Specific VPN MIB	863
	VPN MIB	863
	jnxVpnInfo	864
	jnxVpnTable	864
	jnxVpnIfTable	865
	jnxVpnPwTable	868
	jnxVpnRTTable	873
	VPN Traps	873
Part 3	SNMP Notifications	
Chapter 84	Enterprise-Specific Traps	877
	Antivirus Traps Objects Table	877
	Notifications for the BFD MIB	878
	Juniper Networks Enterprise-Specific BGP Traps	878
	Chassis Traps	879
	SNMPv1 Trap Format	881
	SNMPv2 Trap Format	882
	Configuration Management Notifications	884
	DFC Notification Definitions	885
	Notifications for the Event MIB	886
	jnxTraps	886
	Juniper Networks Enterprise-Specific LDP Traps	887
	LDP Notification Objects and Notification Types	887
	Disabling LDP Traps	889
	Juniper Networks Enterprise-Specific License MIB Notifications	890
	MAC Address Limit Traps	890
	Juniper Networks Enterprise-Specific MIMSTP Traps	891
	Juniper Networks Enterprise-Specific MPLS Traps	892
	Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways	892

	Juniper Networks Enterprise-Specific NAT Traps on MX Series 3D Universal Edge Routers	893
	Service Traps	894
	Juniper Networks Enterprise-Specific SNMP Version 1 Traps	894
	Juniper Networks Enterprise-Specific SNMP Version 2 Traps	902
	Juniper Networks Enterprise-Specific Traps on EX Series Switches	910
	Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers	911
	VPN Traps	911
Chapter 85	Standard Traps	913
	Standard SNMP Version 1 Traps	913
	Standard SNMP Version 2 Traps	916
	SNMP Version 2 MPLS Traps	920
	SNMP Version 2 L3VPN Traps	920
	Standard SNMP Traps on EX Series Ethernet Switches	922
	Unsupported Standard SNMP Traps	922
Part 4	Index	
	Index	929

List of Tables

	About the Documentation	xxxv
	Table 1: Notice Icons	xxxvii
	Table 2: Text and Syntax Conventions	xxxvii
Part 1	Overview	
Chapter 2	SNMP MIBs	9
	Table 3: Standard MIBs Supported on Devices Running Junos OS	9
	Table 4: Enterprise-Specific MIBs and Supported Devices	37
Part 2	Enterprise-Specific MIBs	
Chapter 4	AAA Objects MIB	67
	Table 5: jnxAuthenticateType	68
	Table 6: jnxAccountingType	68
	Table 7: jnxAuthorizationType	69
	Table 8: jnxProvisioningType	69
	Table 9: jnxUserAAAStatTable	70
	Table 10: Access Authentication-Related Traps	70
	Table 11: jnxUserAAAAccessPoolTable	71
Chapter 5	Access Authentication Objects MIB	73
	Table 12: jnxJsFwAuthStats	74
	Table 13: jnxJsAuthTrapVars	74
	Table 14: jnxJsAuthNotifications	75
Chapter 6	Alarm MIB	77
	Table 15: jnxYellowAlarms	78
	Table 16: jnxRedAlarms	78
Chapter 7	Analyzer MIB	81
	Table 17: jnxAnalyzerTable	82
	Table 18: jnxAnalyzerInputTable	82
	Table 19: jnxAnalyzerOutputTable	83
Chapter 8	Antivirus Objects MIB	85
	Table 20: Scan Engine Objects Table	86
	Table 21: Scan Statistics Objects Table	86
	Table 22: Antivirus Traps Objects Table	87
Chapter 9	ATM Class-of-Service MIB	89
	Table 23: jnxCosAtmVcScTable	90
	Table 24: jnxCosAtmVcQstatsTable	92

	Table 25: jnxCosAtmTrunkTable	93
Chapter 10	ATM MIB	95
	Table 26: jnxAtmIfTable	96
	Table 27: jnxAtmVCTable	98
	Table 28: jnxAtmVpTable	101
	Table 29: jnxAtmTrunkTable	102
Chapter 11	BGP4 V2 MIB	105
	Table 30: jnxBgpM2PrefixCountersEntry	106
Chapter 12	Bidirectional Forwarding Detection MIB	107
	Table 31: jnxBfdSessTable	107
	Table 32: Supported Notifications for the BFD MIB	108
Chapter 13	Chassis Cluster MIB	109
	Table 33: Chassis Cluster Switchover Objects	110
	Table 34: Chassis Cluster Trap Objects	110
Chapter 14	Chassis Forwarding MIB	113
	Table 35: jnxFwddProcess	114
Chapter 15	Chassis MIB	115
	Table 36: jnxContainersEntry Objects in the jnxContainersTable of an M5 Router	121
	Table 37: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router	121
	Table 38: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router	122
	Table 39: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router	122
	Table 40: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router	123
	Table 41: jnxContainersEntry Objects in the jnxContainersTable of an MX5 Router	123
	Table 42: jnxContainersEntry Objects in the jnxContainersTable of an MX10 Router	124
	Table 43: jnxContainersEntry Objects in the jnxContainersTable of an MX40 Router	125
	Table 44: jnxContainersEntry Objects in the jnxContainersTable of an MX80 Router	125
	Table 45: jnxContainersEntry Objects in the jnxContainersTable of an MX240 Router	126
	Table 46: jnxContainersEntry Objects in the jnxContainersTable of an MX480 Router	128
	Table 47: jnxContainersEntry Objects in the jnxContainersTable of an MX960 Router	129
	Table 48: jnxContainersEntry Objects in the jnxContainersTable of an MX2010 Router	130
	Table 49: jnxContainersEntry Objects in the jnxContainersTable of an MX2020 Router	130

Table 50: jnxContainersEntry Objects in the jnxContainersTable of a T640 Router	131
Table 51: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router	132
Table 52: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router	132
Table 53: jnxContainersEntry Objects in the jnxContainersTable of a T4000 Router	133
Table 54: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router	135
Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router	138
Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router	146
Table 57: jnxContentsEntry Objects in the jnxContentsTable of a T640 Router	162
Table 58: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router	166
Table 59: jnxContentsEntry Objects in the jnxContentsTable of a T4000 Router	169
Table 60: jnxLEDEntry Objects in the jnxLEDTable of an M20 Router	176
Table 61: jnxLEDEntry Objects in the jnxLEDTable of an MX2010 Router	177
Table 62: jnxLEDEntry Objects in the jnxLEDTable of an MX2020 Router	179
Table 63: jnxLEDEntry Objects in the jnxLEDTable of a T640 Router	182
Table 64: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router	183
Table 65: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router	185
Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router	187
Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router	193
Table 68: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router	203
Table 69: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router	207
Table 70: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router	211
Table 71: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router	218
Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router	219
Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router	227
Table 74: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router	241
Table 75: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router	244
Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router	246
Table 77: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router	252
Table 78: jnxRedundancyEntry Objects in the jnxRedundancyTable of an MX2010 Router	253
Table 79: jnxRedundancyEntry Objects in the jnxRedundancyTable of an MX2020 Router	254

	Table 80: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Router	255
	Table 81: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router	256
	Table 82: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T4000 Router	257
	Table 83: jnxFruContents Objects in the jnxFruTable of an M10 Router	261
	Table 84: JnxFruContents Objects in the jnxFruTable of an M20 Router	264
	Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router	268
	Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router	274
	Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router	280
	Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router	285
	Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router	299
	Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router	324
	Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router	334
	Table 92: SNMP Version 1 Trap Format	349
	Table 93: SNMP Version 2 Trap Format	351
	Table 94: Router Models and Their sysObjectIds	352
Chapter 16	Class-of-Service MIB	371
	Table 95: jnxCosInVQstatEntry	371
Chapter 17	Configuration Management MIB	375
	Table 96: Configuration Management MIB Textual Conventions	376
	Table 97: Configuration Change Management Objects	376
	Table 98: jnxCmCfgChgEventTable	377
	Table 99: Rescue Configuration Change Management Objects	378
Chapter 18	Destination Class Usage MIB	381
	Table 100: jnxDCUsEntry	382
	Table 101: jnxDCUsStatsEntry	382
Chapter 19	DHCP MIB	385
	Table 102: jnxJdhcpLocalServerStatistics	386
	Table 103: jnxJdhcpLocalServerBindings	388
	Table 104: jnxJdhcpLocalServerTrapVars	389
	Table 105: jnxJdhcpLocalServerIfcStats	389
	Table 106: jnxJdhcpRelayStatistics	392
	Table 107: jnxJdhcpRelayBindings	393
	Table 108: jnxJdhcpRelayIfcStats	394
Chapter 20	DHCPv6 MIB	397
	Table 109: jnxJdhcpv6LocalServerStatistics	398
	Table 110: jnxJdhcpv6LocalServerBindings	399
	Table 111: jnxJdhcpv6LocalServerTrapVars	401
	Table 112: jnxJdhcpv6LocalServerIfcStats	401
Chapter 21	Digital Optical Monitoring MIB	405
	Table 113: jnxDomCurrentEntry	406
	Table 114: jnxDomAlarmSet	408
	Table 115: jnxDomAlarmCleared	409

Chapter 22	DNS Objects MIB	411
	Table 116: jnxJsDnsProxyDataObjects	411
Chapter 23	Dynamic Flow Capture MIB	413
	Table 117: jnxDfcCSTable	414
	Table 118: jnxDfcCDTable	417
	Table 119: Supported Notification Variables for the DFC MIB	418
	Table 120: Supported Notification Definitions for the DFC MIB	419
Chapter 24	Ethernet MAC MIB	421
	Table 121: jnxMacStatsTable	421
Chapter 25	Event MIB	423
	Table 122: jnxEventAvTable	423
	Table 123: Supported Notifications for the Event MIB	424
Chapter 27	EX Series MAC Notification MIB	427
	Table 124: jnxMacNotificationGlobalObjects	428
Chapter 28	EX Series SMI MIB	431
	Table 125: jnxExSwitching	431
Chapter 29	EX Series Virtual Chassis MIB	433
	Table 126: jnxVirtualChassisMemberTable	434
Chapter 30	Firewall MIB	435
	Table 127: jnxFirewallsEntry	436
	Table 128: jnxFirewallCounterEntry	436
Chapter 31	Flow Collection Services MIB	439
	Table 129: jnxCollGlobalStats	440
	Table 130: jnxCollPicIfEntry	440
	Table 131: jnxCollFileTable	442
Chapter 33	Interface MIB	447
	Table 132: jnxIfTable	447
	Table 133: ifChassisTable	450
Chapter 34	IP Forward MIB	453
	Table 134: jnxIpCidrRouteTable	454
	Table 135: jnxInetCidrRouteTable	454
Chapter 35	IPsec Generic Flow Monitoring Object MIB	457
	Table 136: IKE Identity Type Textual Conventions	458
	Table 137: IKE Negotiation Mode Text Conventions	458
	Table 138: IKE Negotiations Hash Algorithms	459
	Table 139: IKE Authentication Method	459
	Table 140: Role of Local Endpoint in Negotiations	459
	Table 141: State of Phase 1 IKE Negotiation	460
	Table 142: Diffie-Hellman Group in Negotiations	460
	Table 143: Key Used by IPsec Phase 2 Tunnel	460
	Table 144: Encryption Algorithm in Negotiations	460
	Table 145: Role of Local Endpoint in Negotiations	461

	Table 146: Type of Remote Peer Gateway	461
	Table 147: Role of Local Endpoint in Negotiations	461
	Table 148: Number of IKE Tunnels Currently Active	462
	Table 149: IPsec Phase 1 IKE Tunnel Table	462
	Table 150: IPsec Phase 2 IKE Tunnel Table	466
	Table 151: IPsec Phase 2 Security Association Table	470
Chapter 36	IPsec Monitoring MIB	473
	Table 152: jnxIkeTunnelTable	474
	Table 153: jnxIPSecTunnelTable	476
	Table 154: jnxIPSecSaTable	479
Chapter 37	IPsec VPN Objects MIB	481
	Table 155: jnxJslpSecVpnType	482
	Table 156: jnxJslpSecTunnelTable	482
Chapter 38	IPv4 MIB	485
	Table 157: jnxIpv4AddrTable	485
	Table 158: Standard MIB Groups for IPv6	487
	Table 159: MIB Objects Supported by MX Series and EX Series	502
	Table 160: Supported ICMPv4 Counters	504
	Table 161: Supported ICMPv6 Counters	505
Chapter 39	IPv6 MIB	507
	Table 162: jnxIpv6GlobalStats	507
	Table 163: Count of All Other IPv6 Next Header Types Received	509
	Table 164: jnxIcmpv6GlobalStats	510
	Table 165: Count of All Other ICMPv6 Message Types Received and Transmitted	512
	Table 166: jnxIpv6IfStatsEntry	514
	Table 167: Standard MIB Groups for IPv6	516
	Table 168: MIB Objects Supported by MX Series and EX Series	531
	Table 169: Supported ICMPv4 Counters	533
	Table 170: Supported ICMPv6 Counters	534
Chapter 40	L2ALD MIB	537
	Table 171: jnxL2aldInterfaceTable	538
Chapter 41	L2CP MIB	541
	Table 172: L2CP MIB Objects Supported by the Junos OS	541
Chapter 42	L2TP MIB	543
	Table 173: L2TP Scalar Status and Statistics Group	544
	Table 174: jnxL2tpTunnelGroupStatsTable	545
	Table 175: jnxL2tpTunnelStatsTable	545
	Table 176: jnxL2tpSessionStatsTable	547
	Table 177: jnxL2tpMlpppBundleStatsTable	554
Chapter 43	LDP MIB	557
	Table 178: LDP Notification Objects	558
	Table 179: LDP Notification Types	559
	Table 180: jnxLdpStatsTable	560

Chapter 44	License MIB	563
	Table 181: jnxLicenseInstallEntry	564
	Table 182: jnxLicenseFeatureListEntry	564
	Table 183: License MIB Notifications	565
Chapter 45	Logical Systems MIBs	567
	Table 184: jnxLsysSpZoneObjects	569
	Table 185: jnxLsysSpZoneEntry	569
	Table 186: jnxLsysSpZoneSummary	569
	Table 187: jnxLsysSpSchedulerObjects	571
	Table 188: jnxLsysSpSchedulerEntry	571
	Table 189: jnxLsysSpSchedulerSummary	572
	Table 190: jnxLsysSpZoneObjects	573
	Table 191: jnxLsysSpPolicyEntry	573
	Table 192: jnxLsysSpPolicySummary	574
	Table 193: jnxLsysSpPolicywcntObjects	575
	Table 194: JnxLsysSpPolicywcntEntry	575
	Table 195: jnxLsysSpPolicywcntMIB 2	576
	Table 196: jnxLsysSpFlowgateObjects	577
	Table 197: JnxLsysSpFlowgateEntry	577
	Table 198: jnxLsysSpFlowgateSummary	578
	Table 199: jnxLsysSpFlowsessObjects	579
	Table 200: JnxLsysSpFlowsessEntry	579
	Table 201: jnxLsysSpFlowsessSummary	580
	Table 202: jnxLsysSpAuthentryObjects	581
	Table 203: jnxLsysSpAuthentryEntry	581
	Table 204: jnxLsysSpAuthentrySummary	581
	Table 205: jnxLsysSpNATsrcpoolObjects	583
	Table 206: jnxLsysSpNATsrcpoolEntry	583
	Table 207: jnxLsysSpNATsrcpoolSummary	583
	Table 208: jnxLsysSpNATdstpoolObjects	585
	Table 209: jnxLsysSpNATdstpoolEntry	585
	Table 210: jnxLsysSpNATdstpoolSummary	586
	Table 211: jnxLsysSpNATsrcpatadObjects	587
	Table 212: jnxLsysSpNATsrcpatadEntry	587
	Table 213: jnxLsysSpNATsrcpatadSummary	588
	Table 214: jnxLsysSpNATsrcnopatadObjects	589
	Table 215: jnxLsysSpNATsrcnopatadEntry	589
	Table 216: jnxLsysSpNATsrcnopatadSummary	590
	Table 217: jnxLsysSpNATsrcruleObjects	591
	Table 218: jnxLsysSpNATsrcruleEntry	591
	Table 219: jnxLsysSpNATsrcruleSummary	592
	Table 220: jnxLsysSpNATdstruleObjects	593
	Table 221: jnxLsysSpNATdstruleEntry	593
	Table 222: jnxLsysSpNATdstruleSummary	594
	Table 223: jnxLsysSpNATstaticruleObjects	595
	Table 224: jnxLsysSpNATstaticruleEntry	595
	Table 225: jnxLsysSpNATstaticruleSummary	596
	Table 226: jnxLsysSpNATconebindObjects	597

	Table 227: JnxLsysSpNATconebindEntry	598
	Table 228: jnxLsysSpNATconebindSummary	598
	Table 229: jnxLsysSpNATpoipnumObjects	599
	Table 230: jnxLsysSpNATpoipnumEntry	600
	Table 231: jnxLsysSpNATpoipnumSummary	600
	Table 232: jnxLsysSpCPUObjects	601
	Table 233: JnxLsysSpCPUEntry	602
	Table 234: jnxLsysSpCPSummary	602
	Table 235: jnxLsysSpSPUSummary	603
Chapter 46	MIMSTP MIB	605
	Table 236: jnxMIDot1sJuniperMstTable	606
	Table 237: jnxMIMstMstiBridgeTable	612
	Table 238: jnxMIMstVlanInstanceMappingTable	614
	Table 239: jnxMIMstCistPortTable	615
	Table 240: jnxMIMstMstiPortTable	619
	Table 241: Enterprise-Specific MIMSTP Traps	620
Chapter 48	MPLS MIB	625
	Table 242: mplsInfo	626
	Table 243: mplsTEInfo	626
	Table 244: mplsLsplInfoEntry	627
	Table 245: MPLS Traps	630
Chapter 49	NAT Objects MIB	633
	Table 246: Source NAT Table	634
	Table 247: New Source NAT Table	635
	Table 248: jnxJsNatIfSrcPoolPortTable	637
	Table 249: NAT Rule Hit Table	637
	Table 250: NAT Pool Hit Table	638
	Table 251: NAT Trap Definitions	638
Chapter 50	Network Address Translation Resources—Monitoring MIB	639
	Table 252: Source NAT Table	640
	Table 253: NAT Pool Hit Table	641
	Table 254: NAT Rule Hit Table	641
	Table 255: NAT Trap Definitions	642
Chapter 51	OTN Interface Management MIB	643
	Table 256: jnxOtnAlarmEntry	645
	Table 257: jnxOtnCurrentOdu15minEntry	645
Chapter 52	Packet Forwarding Engine MIB	647
	Table 258: jnxPfeNotifyGlEntry	648
	Table 259: Packet Forwarding Engine Notification Types	649
	Table 260: jnxPfeNotifyTypeEntry	650
Chapter 53	Packet Mirror MIB	651
	Table 261: jnxJsPacketMirrorTrapVars	652
	Table 262: Packet Mirroring-Related Traps	653
Chapter 54	PAE Extension MIB	655

	Table 263: jnxPaeAuthConfigTable	656
	Table 264: jnxStaticMacAuthBypassTable	656
Chapter 55	Passive Monitoring MIB	659
	Table 265: jnxPMFlowEntry	660
Chapter 56	PING MIB	663
	Table 266: jnxPingCtlTable	664
	Table 267: jnxPingsResultsEntry	668
	Table 268: jnxPingProbeHistoryEntry	672
	Table 269: jnxPingLastTestResultEntry	673
Chapter 57	Policy Objects MIB	679
	Table 270: Security Policy Table	680
	Table 271: jnxJsPolicyStatsTable	682
Chapter 58	Power Supply Unit MIB	685
	Table 272: jnxPsuEnvironmentTable	686
	Table 273: jnxPsuTable	686
	Table 274: jnxPsuOutletTable	686
	Table 275: jnxPsuFpcPowerTable	688
Chapter 59	PPP MIB	689
	Table 276: PPP MIB Textual Conventions	690
	Table 277: jnxPppLinkStatusTable	692
	Table 278: jnxPppLinkConfigTable	694
	Table 279: jnxPppNextIfIndex	696
	Table 280: jnxPppIpTable	696
	Table 281: jnxPppIpConfigTable	698
	Table 282: jnxPppOsiTable	699
	Table 283: jnxPppOsiConfigTable	701
	Table 284: jnxPppSessionTable	701
	Table 285: jnxPppMLPppBundleTable	703
	Table 286: jnxPppMLPppNextLinkIfIndex	704
	Table 287: jnxPppMLPppLinkConfigTable	704
	Table 288: jnxPppMLPppNextNetworkIfIndex	706
	Table 289: jnxPppMLPppNetworkConfigTable	707
	Table 290: jnxPppMLPppLinkBindTable	707
	Table 291: jnxPppSummaryCounts	708
	Table 292: jnxPppIpv6Table	714
	Table 293: jnxPppIpv6ConfigTable	715
	Table 294: jnxPppGlobal	715
Chapter 60	PPPoE MIB	717
	Table 295: PPPoE MIB Textual Conventions	718
	Table 296: jnxPPPoENextIfIndex	718
	Table 297: jnxPPPoEIfTable	719
	Table 298: jnxPPPoEIfStatsTable	720
	Table 299: jnxPPPoEIfLockoutTable	722
	Table 300: jnxPPPoESubIfNextIfIndex	723
	Table 301: jnxPPPoESubIfTable	723

	Table 302: jnxPPPoESubIfQueueStatsTable	724
	Table 303: jnxPPPoESummaryCounts	724
	Table 304: jnxPPPoEServiceNameTableTable	726
	Table 305: jnxPPPoEServiceNameTable	726
	Table 306: jnxPPPoEServiceNameAciAriTable	727
Chapter 61	Pseudowire ATM MIB	729
	Table 307: JnxPwAtmCfgEntry	730
	Table 308: jnxpwAtmOutboundNto1Entry	732
	Table 309: jnxpwAtmInboundNto1Entry	734
	Table 310: jnxpwAtmPerfCurrentEntry	735
	Table 311: jnxpwAtmPerfIntervalEntry	737
	Table 312: jnxpwAtmPerf1DayIntervalEntry	738
Chapter 62	Pseudowire TDM MIB	741
	Table 313: jnxpwTDMEntry	743
	Table 314: jnxpwTDMCfgEntry	744
	Table 315: jnxpwTDMPerfCurrentEntry	747
	Table 316: jnxpwTDMPerfIntervalEntry	748
	Table 317: jnxpwTDMPerf1DayIntervalEntry	749
Chapter 63	Real-Time Performance Monitoring MIB	751
	Table 318: jnxRpmResultsSampleTable	752
	Table 319: jnxRpmMeasurementType	752
	Table 320: jnxRpmResultsSummaryTable	754
	Table 321: jnxRpmResultsCalculatedTable	755
	Table 322: jnxRpmHistorySampleTable	756
	Table 323: jnxRpmHistorySummaryTable	756
	Table 324: jnxRpmHistoryCalculatedTable	757
Chapter 64	Reverse Path Forwarding MIB	759
	Table 325: jnxRpfStatsEntry	759
Chapter 65	RMON Events and Alarms MIB	761
	Table 326: jnxRmonAlarmEntry	762
	Table 327: RMON Event and Alarm Traps	763
Chapter 66	RSVP MIB	765
	Table 328: jnxRsvpSessionTable	766
Chapter 67	Security Interface Extension Objects MIB	769
	Table 329: jnxJsIfMonTable	770
Chapter 68	Security Screening Objects MIB	773
	Table 330: jnxJsScreenMonTable	775
	Table 331: jnxJsScreenSweepTable	785
	Table 332: jnxJsScreenNotifications	786
	Table 333: jnxJsScreenTrapVars	787
Chapter 69	Services PIC MIB	789
	Table 334: jnxSpSvcSetTable	790
	Table 335: jnxSpSvcSetSvcTypeTable	791

	Table 336: jnxSpSvcSetIfTable	792
	Table 337: Supported Traps for Services PIC MIB	793
Chapter 70	SNMP IDP MIB	795
	Table 338: jnxJsldpObjects	796
	Table 339: jnxJsldpAttackTable	796
	Table 340: IDP Signature Update and Attack Traps	797
Chapter 71	SONET/SDH Interface Management MIB	799
	Table 341: jnxSonetAlarmTable	799
	Table 342: jnxSonetAlarmInterface Objects in the jnxSonetAlarmTable of an M20 Router	800
Chapter 72	SONET APS MIB	803
	Table 343: apsConfigTable	804
	Table 344: apsStatusTable	805
	Table 345: apsChanConfigEntry	808
	Table 346: apsChanStatusTable	810
Chapter 73	Source Class Usage MIB	813
	Table 347: jnxRpfStatsEntry	814
Chapter 74	SPU Monitoring Objects MIB	815
	Table 348: SPU Monitoring Objects Table	816
	Table 349: Cluster Objects Table	817
Chapter 76	Subscriber MIB	825
	Table 350: Subscriber MIB Textual Conventions	826
	Table 351: Managed Objects for General Subscriber Functions	826
	Table 352: jnxSubscriberTable	827
	Table 353: Managed Objects for Logical System Subscriber Functions	828
	Table 354: jnxSubscriberLogicalSystemTable	829
	Table 355: Managed Objects for Routing Instance Subscriber Functions	829
	Table 356: jnxSubscriberRoutingInstanceTable	830
Chapter 77	System Log MIB	831
	Table 357: jnxSyslogTable	832
	Table 358: Facilities That Generate System Log Messages	833
	Table 359: jnxSyslogAvTable	834
Chapter 78	Traceroute MIB	835
	Table 360: jnxTraceRouteCtlTable	836
Chapter 79	Utility MIB	837
	Table 361: jnxUtilCounter32Entry	838
	Table 362: jnxUtilCounter64Entry	838
	Table 363: jnxUtilIntegerEntry	838
	Table 364: jnxUtilUIntEntry	839
	Table 365: jnxUtilStringEntry	839
Chapter 80	VLAN MIB	841
	Table 366: jnxVlanTable	842
	Table 367: jnxExVlanTable	843

	Table 368: jnxVlanInterfaceTable	843
	Table 369: jnxExVlanInterfaceTable	844
	Table 370: jnxVlanPortGroupTable	844
	Table 371: jnxExVlanPortGroupTable	845
	Table 372: jnxVlanMacListTable	846
Chapter 81	VPLS MIB	849
	Table 373: jnxVplsConfigEntry	850
	Table 374: jnxVplsStatusEntry	852
	Table 375: jnxVplsPwBindEntry	852
	Table 376: jnxVplsBgpADConfigEntry	853
	Table 377: jnxVplsBgpRteTargetEntry	854
	Table 378: jnxVplsBgpVEEntry	856
	Table 379: jnxVplsBgpPwBindEntry	856
Chapter 82	VPN Certificate Objects MIB	859
	Table 380: jnxJsLoadedCaCertTable	860
	Table 381: jnxJsLoadedLocalCertTable	860
Chapter 83	VPN MIB	863
	Table 382: Supported jnxVpnInfo Objects, VPNs, and Circuit Connection Services	864
	Table 383: Supported jnxVpnEntry Objects, VPNs, and Circuit Connection Services	865
	Table 384: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services	866
	Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services	868
	Table 386: Supported jnxVpnRTEntry Objects, VPNs, and Circuit Connection Services	873
	Table 387: Supported VPN Traps, VPNs, and Circuit Connection Services	874
Part 3	SNMP Notifications	
Chapter 84	Enterprise-Specific Traps	877
	Table 388: Antivirus Traps Objects Table	878
	Table 389: Supported Notifications for the BFD MIB	878
	Table 390: SNMP Version 1 Trap Format	882
	Table 391: SNMP Version 2 Trap Format	883
	Table 392: Supported Notification Definitions for the DFC MIB	885
	Table 393: Supported Notifications for the Event MIB	886
	Table 394: LDP Notification Objects	887
	Table 395: LDP Notification Types	888
	Table 396: License MIB Notifications	890
	Table 397: Enterprise-Specific MIMSTP Traps	891
	Table 398: MPLS Traps	892
	Table 399: NAT Trap Definitions	892
	Table 400: NAT Trap Definitions	893
	Table 401: Supported Traps for Services PIC MIB	894

	Table 402: Juniper Networks Enterprise-Specific Supported SNMP Version 1 Traps	895
	Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps	902
	Table 404: Supported VPN Traps, VPNs, and Circuit Connection Services	912
Chapter 85	Standard Traps	913
	Table 405: Standard Supported SNMP Version 1 Traps	913
	Table 406: Standard Supported SNMP Version 2 Traps	917
	Table 407: Unsupported Standard SNMP Traps	923

About the Documentation

- Documentation and Release Notes on page xxxv
- Supported Platforms on page xxxv
- Using the Examples in This Manual on page xxxv
- Documentation Conventions on page xxxvii
- Documentation Feedback on page xxxix
- Requesting Technical Support on page xxxix

Documentation and Release Notes

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

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

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

Supported Platforms

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

- PTX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

Table 1 on page xxxvii 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 xxxvii 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

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

Convention	Description	Examples
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need 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>.

PART 1

Overview

- [Getting Started on page 3](#)
- [SNMP MIBs on page 9](#)
- [SNMP Notifications on page 57](#)

CHAPTER 1

Getting Started

- [Understanding the SNMP Implementation in Junos OS on page 3](#)
- [Loading MIB Files to a Network Management System on page 7](#)

Understanding the SNMP Implementation in Junos OS

SNMP enables the monitoring of network devices from a central location. This topic provides an overview of SNMP and describes how SNMP is implemented in Junos[®] operating system (Junos OS).

This topic includes the following sections:

- [SNMP Architecture on page 3](#)
- [Junos OS SNMP Agent Features on page 6](#)

SNMP Architecture

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

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

Communication between the agent and the manager occurs in one of the following forms:

- **Get**, **GetBulk**, and **GetNext** requests—The manager requests information from the agent; the agent returns the information in a **Get** response message.
- **Set** requests—The manager changes the value of a MIB object controlled by the agent; the agent indicates status in a **Set** response message.
- **Traps** notification—The agent sends traps to notify the manager of significant events that occur on the network device.

This topic contains the following sections:

- [SNMP MIBs on page 4](#)
- [SNMP Traps and Informs on page 4](#)

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 9 in the *SNMP MIBs and Traps Reference* document.

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 25 in the *SNMP MIBs and Traps Reference* document.

SNMP Traps and Informs

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

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

For more information about standard traps supported by Junos OS, see “[Standard SNMP Traps Supported on Devices Running Junos OS](#)” on page 57 in the *SNMP MIBs and Traps Reference* document.

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

For more information about enterprise-specific traps supported by Junos OS, see [“Juniper Networks Enterprise-Specific SNMP Traps” on page 57](#) in the *SNMP MIBs and Traps Reference* document. For information about system logging severity levels for SNMP traps, see [“System Logging Severity Levels for SNMP Traps” on page 5](#).

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*.

SNMP Trap Queuing

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

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

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



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

System Logging Severity Levels for SNMP Traps

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

event with that system logging severity level. For more information about system logging severity levels, see the *Junos OS Administration Library for Routing Devices* document.

For more information about system logging severity levels for standard traps, see [“Standard SNMP Version 1 Traps” on page 913](#) and [“Standard SNMP Version 2 Traps” on page 916](#) in the *SNMP MIBs and Traps Reference* document. For more information about system logging severity levels for enterprise-specific traps, see [“Juniper Networks Enterprise-Specific SNMP Version 1 Traps” on page 894](#) and [“Juniper Networks Enterprise-Specific SNMP Version 2 Traps” on page 902](#) in the *SNMP MIBs and Traps Reference* document.

Junos OS SNMP Agent Features

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

Junos OS supports the following versions of SNMP:

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

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

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

Related Documentation

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

Loading MIB Files to a Network Management System

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

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

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

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

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



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

- a. **mib-SNMPv2-SMI.txt**
- b. **mib-SNMPv2-TC.txt**
- c. **mib-IANAifType-MIB.txt**
- d. **mib-IANA-RTPROTO-MIB.txt**
- e. **mib-rfc1907.txt**
- f. **mib-rfc2011a.txt**

g. `mib-rfc2012a.txt`

h. `mib-rfc2013a.txt`

i. `mib-rfc2863a.txt`

7. Load the remaining standard MIB files.



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

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

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

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



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

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

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)

CHAPTER 2

SNMP MIBs

- Standard SNMP MIBs Supported by Junos OS on page 9
- Juniper Networks Enterprise-Specific MIBs on page 25
- Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36
- SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46
- LLDP Standard MIB Objects Supported on EX Series Devices on page 53

Standard SNMP MIBs Supported by Junos OS

Table 3 on page 9 contains the list of standard SNMP MIBs and RFCs that are supported on various devices running Junos OS. RFCs can be found at <http://www.ietf.org>.



NOTE: In this table, a value of 1 in any of the platform columns (ACX, M, T, J, MX, EX, 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 3: Standard MIBs Supported on Devices Running Junos OS

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

IEEE 802.1ab section 12.1, *Link Layer Discovery Protocol (LLDP) MIB*

0 0 0 0 1 1 0 0 0

EX Series implementation of LLDP MIB supports both IPv4 and IPv6 configuration.

For more information about LLDP MIB objects supported on EX Series devices, see “[LLDP Standard MIB Objects Supported on EX Series Devices](#)” on page 53.

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
IEEE, 802.3ad, <i>Aggregation of Multiple Link Segments</i>	0	1	1	1	1	1	1	1	1
Supported tables and objects:									
<ul style="list-style-type: none"> dot3adAggPortTable, dot3adAggPortListTable, dot3adAggTable, and dot3adAggPortStatsTable 									
NOTE: EX Series switches do not support the dot3adAggPortTable and dot3adAggPortStatsTable.									
<ul style="list-style-type: none"> dot3adAggPortDebugTable (only dot3adAggPortDebugRxState, dot3adAggPortDebugMuxState, dot3adAggPortDebugActorSyncTransitionCount, dot3adAggPortDebugPartnerSyncTransitionCount, dot3adAggPortDebugActorChangeCount, and dot3adAggPortDebugPartnerChangeCount) 									
NOTE: EX Series switches do not support the dot3adAggPortDebugTable.									
<ul style="list-style-type: none"> dot3adTablesLastChanged 									
NOTE: Gigabit Ethernet interfaces on J Series Services Routers do not support the 802.3ad MIB.									
RFC 1155, <i>Structure and Identification of Management Information for TCP/IP-based Internets</i>	1	1	1	1	1	1	1	1	1
RFC 1157, <i>A Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	1	1
RFC 1195, <i>Use of OSI IS-IS for Routing in TCP/IP and Dual Environments</i> (only the objects isisSystem, isisMANAreaAddr, isisAreaAddr, isisSysProtSupp, isisSummAddr, isisCirc, isisCircLevel, isisPacketCount, isisISAdj, isisISAdjAreaAddr, isisAdjIPAddr, isisISAdjProtSupp, isisRa, and isisIPRA are supported)	1	1	1	1	1	1	1	1	1
RFC 1212, <i>Concise MIB Definitions</i>	1	1	1	1	1	1	0	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 1213, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> . Junos OS supports the following areas:	1	1	1	1	1	1	0	0	1
<ul style="list-style-type: none"> MIB II and its SNMP version 2 derivatives, including: <ul style="list-style-type: none"> Statistics counters IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, <i>IP Forwarding Table MIB</i>) SNMP management Interface management SNMPv1 Get, GetNext requests, and version 2 GetBulk request Junos OS-specific secured access list Master configuration keywords Reconfigurations upon SIGHUP 									
RFC 1215, <i>A Convention for Defining Traps for use with the SNMP</i> (only MIB II SNMP version 1 traps and version 2 notifications)	1	1	1	1	1	1	0	0	1
RFC 1406, <i>Definitions of Managed Objects for the DS1 and E1 Interface Types</i> (T1 MIB is supported)	1	1	1	1	0	0	1	0	0
RFC 1407, <i>Definitions of Managed Objects for the DS3/E3 Interface Type</i> (T3 MIB is supported)	0	1	1	1	0	0	0	0	0
RFC 1471, <i>Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol</i> (only pppLink group is supported. The pppLink group consists of the pppLcp1 object and the tables pppLinkStatustable and pppLinkConfigTable).	0	1	0	0	1	0	0	0	0
RFC 1657, <i>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2</i>	1	1	1	1	1	1	0	0	0
RFC 1695, <i>Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2</i>	1	1	1	1	0	0	0	0	0

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 1850, <i>OSPF Version 2 Management Information Base</i> (except for the ospfOriginateNewLsas and ospfRxNewLsas objects, the Host Table, and the traps ospfOriginateLSA , ospfLsdbOverflow , and ospfLsdbApproachingOverflow)	1	1	1	1	1	1	1	0	0
RFC 1901, <i>Introduction to Community-based SNMPv2</i>	1	1	1	1	1	1	1	1	1
RFC 2011, <i>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</i>	1	1	1	1	1	1	0	0	0
RFC 2012, <i>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</i>	1	1	1	1	1	1	1	0	1
RFC 2013, <i>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</i>	1	1	1	1	1	1	1	0	1
RFC 2024, <i>Definitions of Managed Objects for Data Link Switching Using SMIv2</i> (except for the dlswInterface and dlswSdlc object groups; the dlswDirLocateMacTable , dlswDirNBTable , and dlswDirLocateNBTable tables; the dlswCircuitDiscReasonLocal and dlswCircuitDiscReasonRemote tabular objects; and the dlswDirMacCacheNextIndex and dlswDirNBCacheNextIndex scalar objects; read-only access)	0	1	1	1	1	0	0	0	0
RFC 2096, <i>IP Forwarding Table MIB</i> (The ipCidrRouteTable has been extended to include the tunnel name when the next hop is through an RSVP-signaled LSP.) NOTE: RFC 2096 has been replaced by RFC 4292. However, Junos OS currently supports both RFC 2096 and RFC 4292.	1	1	1	1	1	1	0	0	1
RFC 2115, <i>Management Information Base for Frame Relay DTEs Using SMIv2</i> (frDlcmiTable only; frCircuitTable and frErrTable are not supported)	0	1	1	1	1	0	1	0	0
RFC 2233, <i>The Interfaces Group MIB Using SMIv2</i> 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	1	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 2287, <i>Definitions of System-Level Managed Objects for Applications</i> (only the objects sysApplInstallPkgTable , sysApplInstallElmtTable , sysApplElmtRunTable , and sysApplMapTable)	1	1	1	1	1	1	1	0	1
RFC 2465, <i>Management Information Base for IP Version 6: Textual Conventions and General Group</i> (except for IPv6 interface statistics)	1	1	1	1	1	0	1	0	0
RFC 2495, <i>Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types</i> (except for dsx1FarEndConfigTable , dsx1FarEndCurrentTable , dsx1FarEndIntervalTable , dsx1FarEndTotalTable , and dsx1FracTable)	1	1	1	1	0	0	1	0	0
RFC 2515, <i>Definitions of Managed Objects for ATM Management</i> (except atmVpCrossConnectTable , atmVcCrossConnectTable , and aal5VccTable)	1	1	1	1	0	0	0	0	0
RFC 2570, <i>Introduction to Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1	1	0	0	1
RFC 2571, <i>An Architecture for Describing SNMP Management Frameworks</i> (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	1	0	1
RFC 2572, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> (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	1	0	1
RFC 2576, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i> 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	1	0	1
RFC 2578, <i>Structure of Management Information Version 2 (SMIPv2)</i>	1	1	1	1	1	1	0	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 2579, <i>Textual Conventions for SMIv2</i>	1	1	1	1	1	1	0	0	1
RFC 2580, <i>Conformance Statements for SMIv2</i>	1	1	1	1	1	1	0	0	1
RFC 2662, <i>Definitions of Managed Objects for ADSL Lines</i> (J Series Services Routers. All MIB tables, objects, and traps are applicable for the ADSL ATU-R agent.)	0	1	1	1	1	0	1	0	0
RFC 2665, <i>Definitions of Managed Objects for the Ethernet-like Interface Types</i>	1	1	1	1	1	1	1	0	1
<p>NOTE: For M, T and MX Series, the SNMP counters do not count the Ethernet header and frame check sequence (FCS). Therefore, the following four OIDs are not supported:</p> <ul style="list-style-type: none"> • ifInOctets • ifOutOctets • ifHCInOctets • ifHCOctets <p>However, the EX switches adhere to RFC 2665.</p>									
RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> (except row creation, the Set operation, and the object vrrpStatsPacketLengthErrors)	1	1	1	1	1	1	1	0	1
RFC 2790, <i>Host Resources MIB</i>	1	1	1	1	1	1	1	0	1
<ul style="list-style-type: none"> • Only the hrStorageTable. The file systems /, /config, /var, and /tmp always return the same index number. When SNMP restarts, the index numbers for the remaining file systems might change. • Only the objects of the hrSystem and hrSWInstalled groups. 									

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 2819, <i>Remote Network Monitoring Management Information Base</i>	1	1	1	1	1	1	1	0	1
<ul style="list-style-type: none"> etherStatsTable (for Ethernet interfaces only), alarmTable, eventTable, and logTable are supported on all devices running Junos OS. historyControlTable and etherHistoryTable (except etherHistoryUtilization object) are supported only on EX Series switches. 									
RFC 2863, <i>The Interfaces Group MIB</i>	1	1	1	1	1	1	0	0	1
NOTE: RFC 2863 replaces RFC 2233. However, Junos OS supports both RFC 2233 and RFC 2863.									
RFC 2864, <i>The Inverted Stack Table Extension to the Interfaces Group MIB</i>	0	1	1	1	1	0	0	0	1
RFC 2922, <i>The Physical Topology (PTOPO) MIB</i>	0	0	0	0	0	1	1	0	1
Supported objects:									
ptopoConnDiscAlgorithm, ptopoConnAgentNetAddrType, ptopoConnAgentNetAddr, ptopoConnMultiMacSASeen, ptopoConnMultiNetSASeen, ptopoConnIsStatic, ptopoConnLastVerifyTime, ptopoConnRowStatus									
RFC 2925, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i> (only the objects pingCtlTable , pingResultsTable , pingProbeHistoryTable , pingMaxConcurrentRequests , traceRouteCtlTable , traceRouteResultsTable , traceRouteProbeHistoryTable , and traceRouteHopsTable)	1	1	1	1	1	1	1	0	1
RFC 2932, <i>IPv4 Multicast Routing MIB</i>	1	1	1	1	1	1	1	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 2934, <i>Protocol Independent Multicast MIB for IPv4</i>	1	1	1	1	1	1	1	0	0
<p>NOTE: In Junos OS, RFC 2934 is implemented based on a draft version, <i>pimmib.mib</i>, of the now standard RFC.</p> <p>Support for the pimNeighborLoss trap was added in Release 11.4.</p>									
RFC 2981, <i>Event MIB</i>	1	1	1	1	1	0	0	0	0
RFC 3014, <i>Notification Log MIB</i>	1	1	1	1	1	0	0	0	0
RFC 3019, <i>IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol</i>	0	1	1	1	1	0	0	0	1
RFC 3410 <i>Introduction and Applicability Statements for Internet-Standard Management Framework</i>	1	1	1	1	1	1	0	0	1
RFC 3411, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i>	1	1	1	1	1	1	0	0	1
<p>NOTE: RFC 3411 replaces RFC 2571. However, Junos OS supports both RFC 3411 and RFC 2571.</p>									
RFC 3412, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	0	0	1
<p>NOTE: RFC 3412 replaces RFC 2572. However, Junos OS supports both RFC 3412 and RFC 2572.</p>									
RFC 3413, <i>Simple Network Management Protocol (SNMP) Applications</i> (except for the Proxy MIB)	1	1	1	1	1	1	1	0	1
RFC 3414, <i>User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)</i>	1	1	1	1	1	1	0	0	1
RFC 3415, <i>View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	0	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 3416, <i>Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	0	0	1
NOTE: RFC 3416 replaces RFC 1905, which was supported in earlier versions of Junos OS.									
RFC 3417, <i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	1	0	1
RFC 3418, <i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i>	1	1	1	1	1	1	0	0	1
NOTE: RFC 3418 replaces RFC 1907, which was supported in earlier versions of Junos OS.									
RFC 3498, <i>Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures</i> (implemented under the Juniper Networks enterprise branch [jnxExperiment])	0	1	1	1	0	0	0	0	0
RFC 3584 <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i>	1	1	1	1	1	1	0	0	1
RFC 3591 <i>Managed Objects for the Optical Interface Type</i>	0	1	1	1	0	0	0	0	0
optIfOTMnTable (except optIfOTMnOpticalReach, optIfOTMnInterfaceType, and optIfOTMnOrder), optIfOChConfigTable (except optIfOChDirectionality and optIfOChCurrentStatus), optIfOTUkConfigTable (except optIfOTUkTracelIdentifierAccepted, optIfOTUkTIMDetMode, optIfOTUkTIMActEnabled, optIfOTUkTracelIdentifierTransmitted, optIfOTUkDEGThr, optIfOTUkDEGM, optIfOTUkSinkAdaptActive, and optIfOTUkSourceAdaptActive), and optIfODUkConfigTable (except optIfODUkPositionSeqCurrentSize and optIfODUkTtpPresent)									
RFC 3592, <i>Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type</i>	0	1	1	1	1	0	0	0	0
RFC 3621, <i>Power Ethernet MIB</i>	0	0	0	0	0	1	0	0	0

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 3637, <i>Definitions of Managed Objects for the Ethernet WAN Interface Sublayer</i> (except etherWisDeviceTable , etherWisSectionCurrentTable , and etherWisFarEndPathCurrentTable)	0	1	1	1	1	0	0	0	0
RFC 3811, <i>Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management</i>	1	1	1	1	1	0	1	0	0
RFC 3812, <i>Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)</i> (read only access)	1	1	1	1	1	0	0	0	0
<ul style="list-style-type: none"> MPLS tunnels as interfaces are not supported. The following objects in the TunnelResource table are not supported: mplsTunnelResourceMeanRate, mplsTunnelResourceMaxBurstSize, mplsTunnelResourceMeanBurstSize, mplsTunnelResourceExBurstSize, mplsTunnelResourceWeight. mplsTunnelPerfTable and mplsTunnelCRLDPResTable are not supported. mplsTunnelCHopTable is supported on ingress routers only. <p>NOTE: The branch used by the proprietary LDP MIB (ldpmib.mib) conflicts with RFC 3812. ldpmib.mib has been deprecated and replaced by jnx-mpls-ldp.mib.</p>									
RFC 3813, <i>Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)</i> (read-only access). mplsInterfacePerfTable , mplsInSegmentPerfTable , mplsOutSegmentPerfTable , mplsInSegmentMapTable , mplsXCUp , and mplsXCDown are not supported.	1	1	1	1	1	0	1	0	0
RFC 3826, <i>The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model</i>	1	1	1	1	1	1	0	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

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

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

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

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

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

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

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

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
RFC 5643, <i>Management Information Base for OSPFv3</i>	0	1	1	1	1	0	0	0	1
<p>NOTE: Junos OS support for this MIB is read-only.</p> <p>Junos OS does not support the following tables and objects defined in this MIB.</p> <ul style="list-style-type: none"> • ospfv3HostTable • ospfv3CfgrNbrTable • ospfv3ExitOverflowInterval • ospfv3ReferenceBandwidth • ospfv3RestartSupport • ospfv3RestartInterval • ospfv3RestartStrictLsaChecking • ospfv3RestartStatus • ospfv3RestartAge • ospfv3RestartExitReason • ospfv3NotificationEnable • ospfv3StubRouterSupport • ospfv3StubRouterAdvertisement • ospfv3DiscontinuityTime • ospfv3RestartTime • ospfv3AreaNssaTranslatorRole • ospfv3AreaNssaTranslatorState • ospfv3AreaNssaTranslatorStabInterval • ospfv3AreaNssaTranslatorEvents • ospfv3AreaTEEnabled • ospfv3IfMetricValue • ospfv3IfDemandNbrProbe 									
RFC 6527, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)</i> (except row creation, the Set operation, and the objects vrrpv3StatisticsRowDiscontinuityTime and vrrpv3StatisticsPacketLengthErrors)	1	0	0	1	0	0	0	0	0
Internet Assigned Numbers Authority, <i>IANAiftype Textual Convention MIB</i> (referenced by RFC 2233, available at http://www.iana.org/assignments/ianaiftype-mib)	1	1	1	1	1	1	1	0	0

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
Internet draft draft-ietf-atommib-sonetaps-mib-10.txt, <i>Definitions of Managed Objects for SONET Linear APS Architectures</i> (as defined under the Juniper Networks enterprise branch [jnxExperiment] only)	0	1	1	1	1	0	0	0	0
Internet draft draft-ietf-bfd-mib-02.txt, <i>Bidirectional Forwarding Detection Management Information Base</i> (Represented by mib-jnx-bfd-exp.txt and implemented under the Juniper Networks enterprise branch [jnxExperiment]. Read only. Includes bfdSessUp and bfdSessDown traps. Does not support bfdSessPerfTable and bfdSessMapTable .)	1	1	1	1	1	1	0	0	1
Internet draft draft-ietf-idmr-igmp-mib-13.txt, <i>Internet Group Management Protocol (IGMP) MIB</i>	0	1	1	1	1	1	0	0	1
Internet draft draft-reeder-snmpv3-usm-3desede-00.txt, <i>Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode</i>	1	1	1	1	1	1	0	0	1
Internet draft draft-ietf-isis-wg-mib-07.txt, <i>Management Information Base for IS-IS</i> (only isisSAdjTable , isisSAdjAreaAddrTable , isisSAdjIPAddrTable , and isisSAdjProtSuppTable) NOTE: Replaced with RFC 4444, <i>IS-IS MIB</i> in Junos OS Release 11.3 and later.	1	1	1	1	1	1	1	0	0
Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt, <i>MPLS/BGP Virtual Private Network Management Information Base Using SMIv2</i> (only mplsVpnScalars , mplsVpnVrfTable , mplsVpnPerTable , and mplsVpnVrfRouteTargetTable)	0	1	1	1	1	0	0	0	0
Internet draft draft-ietf-ospf-ospfv3-mib-11.txt, <i>Management Information Base for OSPFv3</i> (Represented by mib-jnx-ospfv3mib.txt and implemented under the Juniper Networks enterprise branch [jnxExperiment]. Support for ospfv3NbrTable only. Read only. Object names are prefixed by jnx . For example, jnxOspfv3NbrTable , jnxOspfv3NbrAddressType , and jnxOspfv3NbrPriority .)	0	1	1	1	1	0	0	0	1

Table 3: Standard MIBs Supported on Devices Running Junos OS (*continued*)

MIB/RFC	Platforms								
	ACX	M	T	J	MX	EX	SRX		
							Low-End	Mid-Range	High-End
Internet draft draft-ietf-idmr-pim-mib-09.txt, <i>Protocol Independent Multicast (PIM) MIB</i>	1	1	1	1	1	1	0	0	1
ESO Consortium MIB, which can be found at http://www.snmp.com/eso/	1	1	1	1	1	1	1	0	0
NOTE: The ESO Consortium MIB has been replaced by RFC 3826.									
Internet Draft P2MP MPLS-TE MIB (draft-ietf-mpls-p2mp-te-mib-09.txt) (read-only access) (except <code>mplsTeP2mpTunnelBranchPerfTable</code>).	1	1	1	1	1	0	0	0	0

- Related Documentation**
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Loading MIB Files to a Network Management System on page 7](#)

Juniper Networks Enterprise-Specific MIBs

Junos OS supports the following enterprise-specific MIBs:

- **AAA Objects MIB**—Provides support for monitoring user authentication, authorization, and accounting through the RADIUS, LDAP, SecurID, and local authentication servers. This MIB is currently supported by Junos OS for J Series and SRX Series devices only. For a downloadable version of this MIB, see http://www.Juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-user--aaa.txt. For more information, see “AAA Objects MIB” on page 67.
- **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/junos13.2/topics/reference/mibs/mib-jnx-js-auth.txt. For more information, see “Access Authentication Objects MIB” on page 73.
- **Alarm MIB**—Provides support for alarms from the router. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-alarm.txt. For more information, see “Alarm MIB” on page 77.

- 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/junos13.2/topics/reference/mibs/mib-jnx-analyzer.txt
For more information, see “Analyzer MIB” on page 81.
- 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/junos13.2/topics/reference/mibs/mib-jnx-js-utm-av.txt
For more information, see “Antivirus Objects MIB” on page 85.
- 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/junos13.2/topics/reference/mibs/mib-jnx-atm-cos.txt
For more information, see “ATM Class-of-Service MIB” on page 89.
- 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/junos13.2/topics/reference/mibs/mib-jnx-atm.txt
For more information, see “ATM MIB” on page 95.
- 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/junos13.2/topics/reference/mibs/mib-jnx-bgpmib2.txt
For more information, see “BGP4 V2 MIB” on page 105.
- 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/junos13.2/topics/reference/mibs/mib-jnx-bfd.txt
For more information, see “Bidirectional Forwarding Detection MIB” on page 107.
- 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/junos13.2/topics/reference/mibs/mib-jnx-chas-defines.txt
For more information, see “Chassis MIBs” on page 116.
- Chassis Forwarding MIB—Enables J Series Services Routers to fully support the Junos OS health monitor. This MIB extends the scope of health monitoring to include Junos forwarding process (**fwdd**) components. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-fwdd.txt
For more information, see “Chassis Forwarding MIB” on page 113.

- 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/junos13.2/topics/reference/mibs/mib-jnx-chassis.txt

For more information, see “Chassis MIBs” on page 116.

- 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/junos13.2/topics/reference/mibs/mib-jnx-jsrpd.txt

For more information, see “Chassis Cluster MIB” on page 109.

- 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/junos13.2/topics/reference/mibs/mib-jnx-cos.txt

For more information, see “Class-of-Service MIB” on page 371.

- 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/junos13.2/topics/reference/mibs/mib-jnx-cfgmngmt.txt

For more information, see “Configuration Management MIB” on page 375.

- 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/junos13.2/topics/reference/mibs/mib-jnx-dcu.txt

For more information, see “Destination Class Usage MIB” on page 381.

- 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/junos13.2/topics/reference/mibs/mib-jnx-jdhcp.txt

For more information, see “DHCP MIB” on page 385.

- 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/junos13.2/topics/reference/mibs/mib-jnx-jdhcipv6.txt

For more information, see “DHCPv6 MIB” on page 397.

- 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/junos13.2/topics/reference/mibs/mib-jnx-dom.txt
For more information, see “Digital Optical Monitoring MIB” on page 405.
- 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/junos13.2/topics/reference/mibs/mib-jnx-js-dns.txt
For more information, see “DNS Objects MIB” on page 411.
- 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/junos13.2/topics/reference/mibs/mib-jnx-dfc.txt
For more information, see “Dynamic Flow Capture MIB” on page 413.
- Ethernet MAC MIB—Monitors media access control (MAC) statistics on Gigabit Ethernet intelligent queuing (IQ) interfaces. It collects MAC statistics; for example, **inocets**, **inframes**, **outocets**, 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/junos13.2/topics/reference/mibs/mib-jnx-mac.txt
For more information, see “Ethernet MAC MIB” on page 421.
- 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/junos13.2/topics/reference/mibs/mib-jnx-event.txt
For more information, see “Event MIB” on page 423.
- Experimental MIB—Contains object identifiers for experimental MIBs. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-exp.txt
For more information, see “jnxExperiment” on page 425.
- 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/junos13.2/topics/reference/mibs/mib-jnx-firewall.txt
For more information, see “Firewall MIB” on page 435.
- 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/junos13.2/topics/reference/mibs/mib-jnx-coll.txt
For more information, see “Flow Collection Services MIB” on page 439.
- 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/junos13.2/topics/reference/mibs/mib-jnx-hostresources.txt

For more information, see “Host Resources MIB” on page 445.

- 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/junos13.2/topics/reference/mibs/mib-jnx-hostresources.txt

For more information, see “IDP MIB” on page 795.

- 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/junos13.2/topics/reference/mibs/mib-jnx-if-extensions.txt

For more information, see “Interface MIB” on page 447.

- IP Forward MIB—Extends the standard IP Forwarding Table MIB (RFC 4292) to include CIDR forwarding information. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipforward.txt

For more information, see “IP Forwarding MIB” on page 453.

- 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/junos13.2/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt

For more information, see “IPsec Generic Flow Monitoring Object MIB” on page 457.

- 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/junos13.2/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt

For more information, see “IPSec Monitoring MIB” on page 473.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-ipsec-vpn.txt

For more information, see “IPsec VPN Objects MIB” on page 481.

- 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/junos13.2/topics/reference/mibs/mib-jnx-ipv4.txt
For more information, see “IPv4 MIB” on page 485.
- 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/junos13.2/topics/reference/mibs/mib-jnx-ipv6.txt
For more information, see “IPv6 MIB” on page 507.
- 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/junos13.2/topics/reference/mibs/mib-jnx-l2ald.txt
For more information, see “L2ALD MIB” on page 537.
- 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/junos13.2/topics/reference/mibs/mib-jnx-l2cp-features.txt
For more information, see “L2CP MIB” on page 541.
- L2TP MIB—Provides information about Layer 2 Transport Protocol (L2TP) tunnels and sessions. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2tp.txt
For more information, see “L2TP MIB” on page 543.
- 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/junos13.2/topics/reference/mibs/mib-jnx-ldp.txt
For more information, see “LDP MIB” on page 557.
- 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/junos13.2/topics/reference/mibs/mib-jnx-license.txt
For more information, see “License MIB” on page 563.
- Logical Systems MIBs—Extend SNMP support to logical systems security profile through various MIBs defined under **jnxLsysSecurityProfile**. For a downloadable versions of the MIBs, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt
For more information, see “Logical Systems MIB” on page 567.

- 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/junos13.2/topics/reference/mibs/mib-jnx-mimstp.txt

For more information, see “MIMSTP MIB” on page 605.

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

http://www.juniper.net/techpubs/en_US/junos13.2/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” on page 625.

- 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/junos13.2/topics/reference/mibs/mib-jnx-mpls-ldp.txt



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

For more information, see “MPLS LDP MIB” on page 623.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-nat.txt

For more information, see “NAT Objects MIB” on page 633.

- 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/junos13.2/topics/reference/mibs/mib-jnx-sp-nat.txt

For more information, see “Network Address Translation Resources-Monitoring MIB” on page 639.

- OTN Interface Management MIB—Defines objects for managing Optical Transport Network (OTN) interfaces on devices running Junos OS. For a downloadable version of the MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-otn.txt

For more information, see “OTN Interface Management MIB” on page 643.

- 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/junos13.2/topics/reference/mibs/mib-jnx-pfe.txt

For more information, see “Packet Forwarding Engine MIB” on page 647.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-packet-mirror.txt

For more information, see “Packet Mirror MIB Overview” on page 651.

- 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/junos13.2/topics/reference/mibs/mib-jnx-pae-extension.txt

For more information, see “PAE Extension MIB” on page 655.

- 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/junos13.2/topics/reference/mibs/mib-jnx-pmon.txt

For more information, see “Passive Monitoring MIB” on page 659.

- 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/junos13.2/topics/reference/mibs/mib-jnx-ping.txt

For more information, see “PING MIB” on page 663.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-policy.txt

For more information, see “Policy Objects MIB” on page 679.

- Power Supply Unit MIB—Enables monitoring and managing of the power supply on a device running Junos OS. This MIB is currently supported only on EX Series Ethernet Switches. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-power-supply-unit.txt

For more information, see “Power Supply Unit MIB” on page 685.

- PPP MIB—Provides SNMP support for PPP-related information such as the type of authentication used, interface characteristics, status, and statistics. This MIB is supported on Common Edge PPP process, jpppd. This MIB is currently supported only on M Series and MX Series routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ppp.txt

For more information, see “PPP MIB” on page 689.

- PPPoE MIB—Provides SNMP support for PPPoE-related information such as the type of authentication used, interface characteristics, status, and statistics. This MIB is supported on Common Edge PPPoE process, jpppoed. This MIB is currently supported only on M Series and MX Series routers. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pppoe.txt

For more information, see “PPPoE MIB” on page 717.

- 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/junos13.2/topics/reference/mibs/mib-jnx-pwtdm.txt

For more information, see “Pseudowire TDM MIB” on page 741.

- Pseudowire ATM MIB—Extends the standard Pseudowire MIB, and defines objects used for managing the ATM pseudowires in Juniper products. The enterprise-specific Pseudowire ATM MIB is the Juniper Networks implementation of RFC 5605, *Managed Objects for ATM over Packet Switched Networks (PSNs)*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pwatm.txt

For more information, see “Pseudowire ATM MIB” on page 729.

- 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/junos13.2/topics/reference/mibs/mib-jnx-rpm.txt

For more information, see “Real-Time Performance Monitoring MIB” on page 751.

- 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/junos13.2/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 759.

- 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/junos13.2/topics/reference/mibs/mib-jnx-rmon.txt

For more information, see “RMON Events and Alarms MIB” on page 761.

- 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/junos13.2/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](#)” on page 765.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-if-ext.txt

For more information, see “[Security Interface Extension Objects MIB](#)” on page 769.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-screening.txt

For more information, see “[Security Screening Objects MIB](#)” on page 773.

- 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/junos13.2/topics/reference/mibs/mib-jnx-sp.txt

For more information, see “[Services PIC MIB](#)” on page 789.

- 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/junos13.2/topics/reference/mibs/mib-jnx-sonetaps.txt

For more information, see “[SONET APS MIB](#)” on page 803.

- 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/junos13.2/topics/reference/mibs/mib-jnx-sonetaps.txt

For more information, see “[SONET/SDH Interface Management MIB](#)” on page 799.

- 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/junos13.2/topics/reference/mibs/mib-jnx-scu.txt

For more information, see “[Source Class Usage MIB](#)” on page 813.

- 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/junos13.2/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt
For more information, see “SPU Monitoring Objects MIB” on page 815.
- 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/junos13.2/topics/reference/mibs/mib-jnx-smi.txt
For more information, see “Structure of Management Information MIB” on page 819.
- 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/junos13.2/topics/reference/mibs/mib-jnx-ex-smi.txt
For more information, see “EX Series SMI MIB” on page 431.
- 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/junos13.2/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/junos13.2/topics/reference/mibs/mib-jnx-subscriber.txt
For more information, see “Subscriber MIB” on page 825.
- 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/junos13.2/topics/reference/mibs/mib-jnx-syslog.txt
For more information, see “System Log MIB” on page 831.
- 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/junos13.2/topics/reference/mibs/mib-jnx-traceroute.txt
For more information, see “Traceroute MIB” on page 835.
- Utility MIB—Provides SNMP support for exposing the Junos OS data and has tables that contain information about each type of data, such as integer and string. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-util.txt
For more information, see “Utility MIB” on page 837.
- Virtual Chassis MIB—Contains information about the virtual chassis on the EX Series Ethernet Switches and the MX Series. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-virtualchassis.txt
For more information, see “Virtual Chassis MIBs” on page 433.

- **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/junos13.2/topics/reference/mibs/mib-jnx-vlan.txt

For more information, see “VLAN MIB” on page 841.

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

- **VPLS-Generic-Draft-01-MIB** implemented as **mib-jnx-vpls-generic.txt**
- **VPLS-BGP-Draft-01-MIB** implemented as **mib-jnx-vpls-bgp.txt**
- **VPLS-LDP-Draft-01-MIB** implemented as **mib-jnx-vpls-ldp.txt**

For downloadable versions of these MIBs, see:

- http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-generic.txt
- http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-bgp.txt
- http://www.juniper.net/techpubs/en_US/junos13.2/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/junos13.2/topics/reference/mibs/mib-jnx-js-cert.txt

For more information, see “VPN Certificate Objects MIB” on page 859.

- **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/junos13.2/topics/reference/mibs/mib-jnx-vpn.txt

For more information, see “VPN MIB” on page 863.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [Loading MIB Files to a Network Management System on page 7](#)

Juniper Networks Enterprise-Specific MIBs and Supported Devices

[Table 4 on page 37](#) lists the enterprise-specific MIBs that are supported on various devices running Junos OS.



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



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

Table 4: Enterprise-Specific MIBs and Supported Devices

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
AAA Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-user-aaa.txt	0	1	1	0	0	0	0	0	1	1
Access Authentication Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-auth.txt	0	0	0	0	0	1	0	1	1	1
Alarm MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-alarm.txt	1	1	1	1	1	1	1	1	1	1
Analyzer MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-analyzer.txt	0	0	0	0	1	0	0	0	0	0
Antivirus Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-utm-av.txt	0	0	0	0	0	0	0	1	0	0
ATM Class-of-Service MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-atm-cos.txt	0	1	1	1	0	0	0	1	0	1

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
ATM MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-atm.txt	1	1	1	1	0	0	0	0	0	0
BGP4 V2 MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-bgpmib2.txt	1	1	1	1	1	1	1	1	1	1
Bidirectional Forwarding Detection MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-bfd.txt	1	1	1	1	1	1	1	1	1	1
Chassis Forwarding MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-fwdd.txt	1	0	0	0	0	1	1	1	0	0
Chassis MIBs http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis.txt http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chas-defines.txt	1	1	1	1	1	1	1	1	1	1
Chassis Cluster MIBs http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-jsrpd.txt	0	0	0	0	0	0	0	0	1	1
Class-of-Service MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cos.txt	1	1	1	1	1	1	1	0	0	1
Configuration Management MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cfgmgmt.txt	1	1	1	1	1	1	1	1	1	1
Destination Class Usage MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-dcu.txt	0	1	1	1	0	1	0	0	1	1

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
DHCP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-jdhcp.txt	0	1	1	1	0	0	0	0	0	0
DHCPv6 MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-jdhcpv6.txt	0	1	1	1	0	0	0	0	0	0
Digital Optical Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-dom.txt	1	1	1	1	1	1	0	1	1	1
DNS Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-dns.txt	0	0	0	0	0	0	0	0	1	1
Dynamic Flow Capture MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-dfc.txt	0	1	1	1	0	0	0	0	0	0
Ethernet MAC MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/jnx-mac.txt	0	1	1	1	1	1	0	0	0	1
Event MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-event.txt	1	1	1	1	1	1	1	1	1	1
EX Series MAC Notification MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ex-mac-notification.txt	0	0	0	0	1	0	0	0	0	0
EX Series SMI MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ex-smi.txt	0	0	0	0	1	0	0	0	0	0

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Experimental MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-exp.txt	1	1	1	1	1	1	0	0	0	0
Firewall MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-firewall.txt	1	1	1	1	1	1	1	1	1	1
Flow Collection Services MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-coll.txt	0	1	1	1	0	0	0	0	0	0
Host Resources MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-hostresources.txt	1	1	1	1	1	1	0	1	1	1
Interface MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-if-extensions.txt	1	1	1	1	1	1	1	1	1	1
IP Forward MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipforward.txt	1	1	1	1	1	1	1	1	1	1
IPsec Generic Flow Monitoring Object MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipsec-flow-mon.txt	0	0	0	1	0	0	0	1	1	1
IPsec Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt	0	1	1	1	0	1	0	0	0	0
IPsec VPN Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-ipsec-vpn.txt	0	0	0	1	0	0	0	1	1	1

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
IPv4 MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipv4.txt	1	1	1	1	1	1	1	1	1	1
IPv6 and ICMPv6 MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ipv6.txt	0	1	1	1	1	0	1	1	1	1
L2ALD MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2ald.txt	0	0	1	0	1	0	0	0	0	0
L2CP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2cp-features.txt	0	0	0	0	1	0	0	0	0	0
L2TP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2tp.txt	0	1	1	0	0	0	0	0	0	0
LDP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ldp.txt	1	1	1	1	0	0	1	0	0	1
License MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-license.txt	0	1	1	1	0	0	0	1	1	1
Logical Systems MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt	0	0	0	0	0	0	0	0	1	1
MIMSTP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mimstp.txt	0	0	1	0	1	0	0	0	0	0

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
MPLS LDP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls-ldp.txt	1	1	1	1	1	1	1	0	0	0
MPLS MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls.txt	1	1	1	1	1	1	1	0	0	1
NAT Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-nat.txt	0	0	0	0	0	1	0	1	1	1
NAT Resources-Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sp-nat.txt	0	1	1	1	0	0	0	0	0	0
OTN Interface Management MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-otn.txt	0	1	1	1	0	0	0	0	0	0
Packet Forwarding Engine MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pfe.txt	1	1	1	1	0	1	1	1	1	1
Packet Mirror MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-packet-mirror.txt	0	0	0	0	1	0	0	0	0	0
PAE Extension MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pae-extension.txt	0	0	0	0	1	0	0	0	0	0
Passive Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pmon.txt	0	1	1	1	0	0	0	0	0	0

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
Ping MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ping.txt	1	1	1	1	1	1	0	1	1	1
Policy Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-policy.txt	0	0	0	0	0	1	0	1	1	1
Power Supply Unit MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-power-supply-unit.txt	0	0	0	0	1	0	1	0	0	0
PPP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ppp.txt	0	1	1	0	0	0	0	0	0	0
PPPoE MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pppoe.txt	0	1	1	0	0	0	0	0	0	0
Pseudowire ATM MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pwatm.txt	0	1	0	0	1	0	0	0	0	0
Pseudowire TDM MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pwtdm.txt	1	1	1	1	0	0	0	0	0	0
Real-Time Performance Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rpm.txt	0	1	1	1	1	1	0	1	0	0
Reverse-Path-Forwarding MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rpf.txt	1	1	1	1	1	1	1	1	1	1

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
RMON Events and Alarms MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rmon.txt	1	1	1	1	1	1	1	1	1	1
RSVP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rsvp.txt	1	1	1	1	1	0	1	0	0	0
Security Interface Extension Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-if-ext.txt	0	0	0	0	0	1	0	1	1	1
Security Screening Objects MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-screening.txt	0	0	0	0	0	0	0	0	0	1
Services PIC MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sp.txt	0	1	1	1	0	0	0	0	0	0
SNMP IDP MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-idp.txt	0	0	0	0	0	0	0	1	1	1
SONET APS MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sonetaps.txt	0	1	1	1	0	0	0	0	0	0
SONET/SDH Interface Management MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sonet.txt	0	1	1	1	0	0	0	0	0	0
Source Class Usage MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-scu.txt	0	1	1	1	0	0	0	0	0	1

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
SPU Monitoring MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-spu-monitoring.txt	0	0	0	0	0	0	0	1	1	1
Structure of Management Information MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-smi.txt	1	1	1	1	1	1	0	1	1	1
Subscriber MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-subscriber.txt	1	0	1	0	0	0	0	0	0	0
System Log MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-syslog.txt	0	1	1	1	1	1	1	1	1	1
Traceroute MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-traceroute.txt	0	1	1	1	1	1	0	1	1	1
Utility MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-util.txt	0	1	1	1	1	1	0	1	1	1
Virtual Chassis MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-virtualchassis.txt	0	0	0	0	1	1	0	0	0	0
VLAN MIB http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vlan.txt	0	0	0	0	1	0	0	0	0	0

Table 4: Enterprise-Specific MIBs and Supported Devices (*continued*)

Enterprise-Specific MIB	Platforms									
	ACX	M	T	J	MX	EX	PTX	SRX		
								Low-End	Mid-Range	High-End
VPLS MIBs	0	1	1	1	1	0	0	0	0	0
<ul style="list-style-type: none"> • http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-generic.txt • http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-ldp.txt • http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-bgp.txt 										
VPN Certificate Objects MIB	0	0	0	0	0	1	0	1	1	1
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-cert.txt										
VPN MIB	1	1	1	1	1	1	0	0	0	0
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpn.txt										

- Related Documentation**
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
 - [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Loading MIB Files to a Network Management System on page 7](#)

SNMP MIB Objects Supported by Junos OS for the Set Operation

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

Object Name	Object Identifier
RFC 1907	
sysContact	1.3.6.1.2.1.1.4
sysName	1.3.6.1.2.1.1.5
sysLocation	1.3.6.1.2.1.1.6
snmpEnableAuthenTraps	1.3.6.1.2.1.11.30

Object Name	Object Identifier
RFC 2819a	
alarmInterval	1.3.6.1.2.1.16.3.1.1.2
alarmVariable	1.3.6.1.2.1.16.3.1.1.2
alarmSampleType	1.3.6.1.2.1.16.3.1.1.4
alarmStartupAlarm	1.3.6.1.2.1.16.3.1.1.6
alarmRisingThreshold	1.3.6.1.2.1.16.3.1.1.7
alarmFallingThreshold	1.3.6.1.2.1.16.3.1.1.8
alarmRisingEventIndex	1.3.6.1.2.1.16.3.1.1.9
alarmFallingEventIndex	1.3.6.1.2.1.16.3.1.1.10
alarmOwner	1.3.6.1.2.1.16.3.1.1.11
alarmStatus	1.3.6.1.2.1.16.3.1.1.12
eventDescription	1.3.6.1.2.1.16.9.1.1.2
eventType	1.3.6.1.2.1.16.9.1.1.3
eventCommunity	1.3.6.1.2.1.16.9.1.1.4
eventOwner	1.3.6.1.2.1.16.9.1.1.6
eventStatus	1.3.6.1.2.1.16.9.1.1.7
RFC 2925a	
pingMaxConcurrentRequests	1.3.6.1.2.1.80.1.1
pingCtlTargetAddressType	1.3.6.1.2.1.80.1.2.1.3
pingCtlTargetAddress	1.3.6.1.2.1.80.1.2.1.4
pingCtlDataSize	1.3.6.1.2.1.80.1.2.1.5
pingCtlTimeOut	1.3.6.1.2.1.80.1.2.1.6
pingCtlProbeCount	1.3.6.1.2.1.80.1.2.1.7
pingCtlAdminStatus	1.3.6.1.2.1.80.1.2.1.8

Object Name	Object Identifier
pingCtlDataFill	1.3.6.1.2.1.80.1.2.1.9
pingCtlFrequency	1.3.6.1.2.1.80.1.2.1.10
pingCtlMaxRows	1.3.6.1.2.1.80.1.2.1.11
pingCtlStorageType	1.3.6.1.2.1.80.1.2.1.12
pingCtlTrapGeneration	1.3.6.1.2.1.80.1.2.1.13
pingCtlTrapProbeFailureFilter	1.3.6.1.2.1.80.1.2.1.14
pingCtlTrapTestFailureFilter	1.3.6.1.2.1.80.1.2.1.15
pingCtlType	1.3.6.1.2.1.80.1.2.1.16
pingCtlDescr	1.3.6.1.2.1.80.1.2.1.17
pingCtlSourceAddressType	1.3.6.1.2.1.80.1.2.1.18
pingCtlSourceAddress	1.3.6.1.2.1.80.1.2.1.19
pingCtlIfIndex	1.3.6.1.2.1.80.1.2.1.20
pingCtlByPassRouteTable	1.3.6.1.2.1.80.1.2.1.21
pingCtlDSField	1.3.6.1.2.1.80.1.2.1.22
pingCtlRowStatus	1.3.6.1.2.1.80.1.2.1.23
RFC 2925B	
traceRouteMaxConcurrentRequests	1.3.6.1.2.1.81.1.1
traceRouteCtlTargetAddressType	1.3.6.1.2.1.81.1.2.1.3
traceRouteCtlTargetAddress	1.3.6.1.2.1.81.1.2.1.4
traceRouteCtlByPassRouteTable	1.3.6.1.2.1.81.1.2.1.5
traceRouteCtlDataSize	1.3.6.1.2.1.81.1.2.1.6
traceRouteCtlTimeOut	1.3.6.1.2.1.81.1.2.1.7
traceRouteCtlProbesPerHop	1.3.6.1.2.1.81.1.2.1.8
traceRouteCtlPort	1.3.6.1.2.1.81.1.2.1.9

Object Name	Object Identifier
traceRouteCtlMaxTtl	1.3.6.1.2.1.81.1.2.1.10
traceRouteCtlDSField	1.3.6.1.2.1.81.1.2.1.11
traceRouteCtlSourceAddressType	1.3.6.1.2.1.81.1.2.1.12
traceRouteCtlSourceAddress	1.3.6.1.2.1.81.1.2.1.13
traceRouteCtlIfIndex	1.3.6.1.2.1.81.1.2.1.14
traceRouteCtlMiscOptions	1.3.6.1.2.1.81.1.2.1.15
traceRouteCtlMaxFailure	1.3.6.1.2.1.81.1.2.1.16
traceRouteCtlDontFragment	1.3.6.1.2.1.81.1.2.1.17
traceRouteCtlInitialTtl	1.3.6.1.2.1.81.1.2.1.18
traceRouteCtlFrequency	1.3.6.1.2.1.81.1.2.1.19
traceRouteCtlStorageType	1.3.6.1.2.1.81.1.2.1.20
traceRouteCtlAdminStatus	1.3.6.1.2.1.81.1.2.1.21
traceRouteCtlDescr	1.3.6.1.2.1.81.1.2.1.22
traceRouteCtlMaxRows	1.3.6.1.2.1.81.1.2.1.23
traceRouteCtlTrapGeneration	1.3.6.1.2.1.81.1.2.1.24
traceRouteCtlCreateHopEntries	1.3.6.1.2.1.81.1.2.1.25
traceRouteCtlType	1.3.6.1.2.1.81.1.2.1.26
traceRouteCtlRowStatus	1.3.6.1.2.1.81.1.2.1.27
Enterprise-Specific PING MIB	
jnxPingCtlIfName	1.3.6.1.4.1.2636.3.7.1.2.1.3
jnxPingCtlRoutingIfIndex	1.3.6.1.4.1.2636.3.7.1.2.1.4
jnxPingCtlRoutingIfName	1.3.6.1.4.1.2636.3.7.1.2.1.5
jnxPingCtlRoutingInstanceName	1.3.6.1.4.1.2636.3.7.1.2.1.6
jnxPingCtlRttThreshold	1.3.6.1.4.1.2636.3.7.1.2.1.7

Object Name	Object Identifier
jnxPingCtlRttStdDevThreshold	1.3.6.1.4.1.2636.3.71.2.1.8
jnxPingCtlRttJitterThreshold	1.3.6.1.4.1.2636.3.71.2.1.9
jnxPingCtlEgressTimeThreshold	1.3.6.1.4.1.2636.3.71.2.1.10
jnxPingCtlEgressStdDevThreshold	1.3.6.1.4.1.2636.3.71.2.1.11
jnxPingCtlEgressJitterThreshold	1.3.6.1.4.1.2636.3.71.2.1.12
jnxPingCtlIngressTimeThreshold	1.3.6.1.4.1.2636.3.71.2.1.13
jnxPingCtlIngressStdDevThreshold	1.3.6.1.4.1.2636.3.71.2.1.14
jnxPingCtlIngressJitterThreshold	1.3.6.1.4.1.2636.3.71.2.1.15
jnxPingTrapGeneration	1.3.6.1.4.1.2636.3.71.2.1.16
Enterprise-Specific Traceroute MIB	
jnxTRCtlIfName	1.3.6.1.4.1.2636.3.8.1.2.1.3
jnxTRCtlRoutingInstanceName	1.3.6.1.4.1.2636.3.8.1.2.1.4
RFC 3413 Target MIB	
snmpTargetSpinLock	1.3.6.1.6.3.12.1.1
snmpTargetAddrTDomain	1.3.6.1.6.3.12.1.2.1.2
snmpTargetAddrTAddress	1.3.6.1.6.3.12.1.2.1.3
snmpTargetAddrTimeout	1.3.6.1.6.3.12.1.2.1.4
snmpTargetAddrRetryCount	1.3.6.1.6.3.12.1.2.1.5
snmpTargetAddrTagList	1.3.6.1.6.3.12.1.2.1.6
snmpTargetAddrParams	1.3.6.1.6.3.12.1.2.1.7
snmpTargetAddrStorageType	1.3.6.1.6.3.12.1.2.1.8
snmpTargetAddrRowStatus	1.3.6.1.6.3.12.1.2.1.9
snmpTargetParamsMPModel	1.3.6.1.6.3.12.1.3.1.2
snmpTargetParamsSecurityModel	1.3.6.1.6.3.12.1.3.1.3

Object Name	Object Identifier
snmpTargetParamsSecurityLevel	1.3.6.1.6.3.12.1.3.1.4
snmpTargetParamsSecurityName	1.3.6.1.6.3.12.1.3.1.5
snmpTargetParamsStorageType	1.3.6.1.6.3.12.1.3.1.6
snmpTargetParamsRowStatus	1.3.6.1.6.3.12.1.3.1.7
RFC 3413 Notify MIB	
snmpNotifyTag	1.3.6.1.6.3.13.1.1.1.2
snmpNotifyType	1.3.6.1.6.3.13.1.1.1.3
snmpNotifyStorageType	1.3.6.1.6.3.13.1.1.1.4
snmpNotifyRowStatus	1.3.6.1.6.3.13.1.1.1.5
snmpNotifyFilterProfileName	1.3.6.1.6.3.13.1.2.1.1
snmpNotifyFilterProfileStorType	1.3.6.1.6.3.13.1.2.1.2
snmpNotifyFilterProfileRowStatus	1.3.6.1.6.3.13.1.2.1.3
snmpNotifyFilterMask	1.3.6.1.6.3.13.1.3.1.2
snmpNotifyFilterType	1.3.6.1.6.3.13.1.3.1.3
snmpNotifyFilterStorageType	1.3.6.1.6.3.13.1.3.1.4
snmpNotifyFilterRowStatus	1.3.6.1.6.3.13.1.3.1.5
RFC 2574	
usmUserSpinLock	1.3.6.1.6.3.15.1.2.1
usmUserCloneFrom	1.3.6.1.6.3.15.1.2.2.1.4
usmUserAuthProtocol	1.3.6.1.6.3.15.1.2.2.1.5
usmUserAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.6
usmUserOwnAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.7
usmUserPrivProtocol	1.3.6.1.6.3.15.1.2.2.1.8
usmUserPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.9

Object Name	Object Identifier
usmUserOwnPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.10
usmUserPublic	1.3.6.1.6.3.15.1.2.2.1.11
usmUserStorageType	1.3.6.1.6.3.15.1.2.2.1.12
usmUserStatus	1.3.6.1.6.3.15.1.2.2.1.13
RFC 2575	
vacmGroupName	1.3.6.1.6.3.16.1.2.1.3
vacmSecurityToGroupStorageType	1.3.6.1.6.3.16.1.2.1.4
vacmSecurityToGroupStatus	1.3.6.1.6.3.16.1.2.1.5
vacmAccessContextMatch	1.3.6.1.6.3.16.1.4.1.4
vacmAccessReadViewName	1.3.6.1.6.3.16.1.4.1.5
vacmAccessWriteViewName	1.3.6.1.6.3.16.1.4.1.6
vacmAccessNotifyViewName	1.3.6.1.6.3.16.1.4.1.7
vacmAccessStorageType	1.3.6.1.6.3.16.1.4.1.8
vacmAccessStatus	1.3.6.1.6.3.16.1.4.1.9
vacmViewSpinLock	1.3.6.1.6.3.16.1.5.1
vacmViewTreeFamilyMask	1.3.6.1.6.3.16.1.5.2.1.3
vacmViewTreeFamilyType	1.3.6.1.6.3.16.1.5.2.1.4
vacmViewTreeFamilyStorageType	1.3.6.1.6.3.16.1.5.2.1.5
vacmViewTreeFamilyStatus	1.3.6.1.6.3.16.1.5.2.1.6
RFC 2576	
snmpCommunityName	1.3.6.1.6.3.18.1.1.1.2
snmpCommunitySecurityName	1.3.6.1.6.3.18.1.1.1.3
snmpCommunityContextEngineID	1.3.6.1.6.3.18.1.1.1.4
snmpCommunityContextName	1.3.6.1.6.3.18.1.1.1.5

Object Name	Object Identifier
snmpCommunityTransportTag	1.3.6.1.6.3.18.1.1.1.6
snmpCommunityStorageType	1.3.6.1.6.3.18.1.1.1.7
snmpCommunityStatus	1.3.6.1.6.3.18.1.1.1.8
RFC 2576	
snmpTargetAddrMask	1.3.6.1.6.3.18.1.2.1.1
snmpTargetAddrMMS	1.3.6.1.6.3.18.1.2.1.2

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)

LLDP Standard MIB Objects Supported on EX Series Devices

Juniper Networks EX Series devices support IEEE 802.1ab section 12.1, *Link Layer Discovery Protocol (LLDP) MIB*. LLDP MIB implementation on EX Series devices supports both IPv4 and IPv6 configurations.

LLDP MIB implementation on EX Series devices supports only the following tables and objects.

LLDP MIB groups supported on EX Series devices are:

- lldpConfigRxGroup
- lldpConfigTxGroup
- lldpStatsRxGroup
- lldpStatsTxGroup
- lldpLocSystemsGroup
- lldpRemSystemsGroup
- lldpNotificationsGroup

The following LLDP configuration group objects are supported:

- lldpMessageTxInterval
- lldpMessageTxHoldMultiplier
- lldpTxDelay
- lldpPortConfigTable

- lldpPortConfigEntry
- lldpPortConfigPortNum
- lldpPortConfigAdminStatus
- lldpPortConfigNotificationEnable
- lldpPortConfigTLVsTxEnable
- lldpConfigManAddrTable
- lldpConfigManAddrEntry
- lldpConfigManAddrPortsTxEnable

The following LLDP statistics group objects are supported:

- lldpStatsRemTablesLastChangeTime
- lldpStatsRemTablesInserts
- lldpStatsRemTablesDeletes
- lldpStatsRemTablesDrops
- lldpStatsRemTablesAgeouts
- lldpStatsTxPortTable
- lldpStatsTxPortEntry
- lldpStatsTxPortNum
- lldpStatsTxPortFramesTotal
- lldpStatsRxPortTable
- lldpStatsRxPortEntry
- lldpStatsRxPortNum
- lldpStatsRxPortFramesTotal

The following LLDP local system group objects are supported:

- lldpLocChassisIdSubtype
- lldpLocChassisId
- lldpLocSysName
- lldpLocSysDesc
- lldpLocSysCapSupported
- lldpLocSysCapEnabled
- lldpLocPortTable
- lldpLocPortEntry
- lldpLocPortNum

- lldpLocPortIdSubtype
- lldpLocPortId
- lldpLocPortDesc
- lldpLocManAddrTable
- lldpLocManAddrEntry
- lldpLocManAddrSubtype
- lldpLocManAddr
- lldpLocManAddrLen
- lldpLocManAddrIfSubtype
- lldpLocManAddrIfId
- lldpLocManAddrOID

The following LLDP remote system group objects are supported:

- lldpRemTable
- lldpRemEntry
- lldpRemTimeMark
- lldpRemLocalPortNum
- lldpRemIndex
- lldpRemChassisIdSubtype
- lldpRemChassisId
- lldpRemPortIdSubtype
- lldpRemPortId
- lldpRemPortDesc
- lldpRemSysName
- lldpRemSysDesc
- lldpRemSysCapSupported
- lldpRemSysCapEnabled
- lldpRemManAddrTable
- lldpRemManAddrEntry
- lldpRemManAddrSubtype
- lldpRemManAddr
- lldpRemManAddrIfSubtype
- lldpRemManAddrIfId
- lldpRemManAddrOID

- lldpRemUnknownTLVTable
- lldpRemUnknownTLVEntry
- lldpRemUnknownTLVType
- lldpRemUnknownTLVInfo
- lldpRemOrgDefInfoTable
- lldpRemOrgDefInfoEntry
- lldpRemOrgDefInfoOUI
- lldpRemOrgDefInfoSubtype
- lldpRemOrgDefInfoIndex
- lldpRemOrgDefInfo

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)

CHAPTER 3

SNMP Notifications

- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
- [Managing Traps and Informs on page 58](#)

Standard SNMP Traps Supported on Devices Running Junos OS

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



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

- [Standard SNMP Version 1 Traps on page 913](#)
- [Standard SNMP Version 2 Traps on page 916](#)
- [Standard SNMP Traps on EX Series Ethernet Switches on page 922](#)
- [Unsupported Standard SNMP Traps on page 922](#)

Related Documentation

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

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 894](#)
- [Juniper Networks Enterprise-Specific SNMP Version 2 Traps on page 902](#)
- [Juniper Networks Enterprise-Specific BGP Traps on page 878](#)
- [Juniper Networks Enterprise-Specific LDP Traps on page 558](#)
- [Juniper Networks Enterprise-Specific License MIB Notifications on page 565](#)
- [Juniper Networks Enterprise-Specific MIMSTP Traps on page 620](#)
- [Juniper Networks Enterprise-Specific MPLS Traps on page 630](#)



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

- [Juniper Networks Enterprise-Specific Traps on EX Series Switches on page 910](#)
- [Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers on page 911](#)

Related Documentation

- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Managing Traps and Informs

The following sections contain a few tips on managing SNMP notifications:

- [Generating Traps Based on SysLog Events on page 58](#)
- [Filtering Traps Based on the Trap Category on page 59](#)
- [Filtering Traps Based on the Object Identifier on page 59](#)

Generating Traps Based on SysLog Events

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 is no corresponding trap, into a trap. If you are using network management

system traps rather than system log messages to monitor your network, you can use this feature to ensure that you are notified of all the major events.

To configure a policy that raises a trap on receipt of an event, include the following statements at the `[edit event-options policy policy-name]` hierarchy level:

```
[edit event-options policy policy-name]
events [ events ];
then {
    raise-trap;
}
```

The following example shows the sample configuration for raising a trap for the event `ui_mgd_terminate`:

Generating Traps Based on SysLog Events	<pre>[edit event-options policy p1] events ui_mgd_terminate; then { raise-trap; }</pre>
--	---

Filtering Traps Based on the Trap Category

SNMP traps are categorized into many categories. The Junos OS provides a configuration option, **categories** at the `[edit snmp trap-group trap-group]` hierarchy level, that enables you to specify categories of traps that you want to receive on a particular host. You can use this option when you want to monitor only specific modules of the Junos OS.

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

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

Filtering Traps Based on the Object Identifier

The Junos OS also provides a more advanced filter option that enables you to filter out specific traps based on their object identifiers. You can use the **notify-filter** option to filter out a specific trap or a group of traps.

The following example shows the sample configuration for excluding Juniper Networks enterprise-specific configuration management traps (note that the SNMPv3 configuration also supports filtering of SNMPv1 and SNMPv2 traps as is shown in the following example):

```
[edit snmp]
```

```
v3 {
  vacm {
    security-to-group {
      security-model v2c {
        security-name sn_v2c_trap {
          group gr_v2c_trap;
        }
      }
    }
  }
  access {
    group gr_v2c_trap {
      default-context-prefix {
        security-model v2c {
          security-level none {
            read-view all;
            notify-view all;
          }
        }
      }
    }
  }
}
target-address TA_v2c_trap {
  address 10.209.196.166;
  port 9001;
  tag-list tg1;
  target-parameters TP_v2c_trap;
}
target-parameters TP_v2c_trap {
  parameters {
    message-processing-model v2c;
    security-model v2c;
    security-level none;
    security-name sn_v2c_trap;
  }
  notify-filter nf1;
}
notify v2c_notify {
  type trap;
  tag tg1;
}
notify-filter nf1 {
  oid .1.3.6.1.4.1.2636.4.5 exclude;
  oid .1 include;
}
snmp-community index1 {
  community-name "$9$tDLl01h7Nbw2axN"; ## SECRET-DATA
  security-name sn_v2c_trap;
  tag tg1;
}
view all {
  oid .1 include;
}
}
```

**Related
Documentation**

PART 2

Enterprise-Specific MIBs

- [AAA Objects MIB on page 67](#)
- [Access Authentication Objects MIB on page 73](#)
- [Alarm MIB on page 77](#)
- [Analyzer MIB on page 81](#)
- [Antivirus Objects MIB on page 85](#)
- [ATM Class-of-Service MIB on page 89](#)
- [ATM MIB on page 95](#)
- [BGP4 V2 MIB on page 105](#)
- [Bidirectional Forwarding Detection MIB on page 107](#)
- [Chassis Cluster MIB on page 109](#)
- [Chassis Forwarding MIB on page 113](#)
- [Chassis MIB on page 115](#)
- [Class-of-Service MIB on page 371](#)
- [Configuration Management MIB on page 375](#)
- [Destination Class Usage MIB on page 381](#)
- [DHCP MIB on page 385](#)
- [DHCPv6 MIB on page 397](#)
- [Digital Optical Monitoring MIB on page 405](#)
- [DNS Objects MIB on page 411](#)
- [Dynamic Flow Capture MIB on page 413](#)
- [Ethernet MAC MIB on page 421](#)
- [Event MIB on page 423](#)
- [Experiment MIB on page 425](#)
- [EX Series MAC Notification MIB on page 427](#)
- [EX Series SMI MIB on page 431](#)
- [EX Series Virtual Chassis MIB on page 433](#)
- [Firewall MIB on page 435](#)
- [Flow Collection Services MIB on page 439](#)
- [Host Resources MIB on page 445](#)

- [Interface MIB on page 447](#)
- [IP Forward MIB on page 453](#)
- [IPsec Generic Flow Monitoring Object MIB on page 457](#)
- [IPsec Monitoring MIB on page 473](#)
- [IPsec VPN Objects MIB on page 481](#)
- [IPv4 MIB on page 485](#)
- [IPv6 MIB on page 507](#)
- [L2ALD MIB on page 537](#)
- [L2CP MIB on page 541](#)
- [L2TP MIB on page 543](#)
- [LDP MIB on page 557](#)
- [License MIB on page 563](#)
- [Logical Systems MIBs on page 567](#)
- [MIMSTP MIB on page 605](#)
- [MPLS LDP MIB on page 623](#)
- [MPLS MIB on page 625](#)
- [NAT Objects MIB on page 633](#)
- [Network Address Translation Resources—Monitoring MIB on page 639](#)
- [OTN Interface Management MIB on page 643](#)
- [Packet Forwarding Engine MIB on page 647](#)
- [Packet Mirror MIB on page 651](#)
- [PAE Extension MIB on page 655](#)
- [Passive Monitoring MIB on page 659](#)
- [PING MIB on page 663](#)
- [Policy Objects MIB on page 679](#)
- [Power Supply Unit MIB on page 685](#)
- [PPP MIB on page 689](#)
- [PPPoE MIB on page 717](#)
- [Pseudowire ATM MIB on page 729](#)
- [Pseudowire TDM MIB on page 741](#)
- [Real-Time Performance Monitoring MIB on page 751](#)
- [Reverse Path Forwarding MIB on page 759](#)
- [RMON Events and Alarms MIB on page 761](#)
- [RSVP MIB on page 765](#)
- [Security Interface Extension Objects MIB on page 769](#)
- [Security Screening Objects MIB on page 773](#)
- [Services PIC MIB on page 789](#)

- [SNMP IDP MIB on page 795](#)
- [SONET/SDH Interface Management MIB on page 799](#)
- [SONET APS MIB on page 803](#)
- [Source Class Usage MIB on page 813](#)
- [SPU Monitoring Objects MIB on page 815](#)
- [Structure of Management Information MIB on page 819](#)
- [Subscriber MIB on page 825](#)
- [System Log MIB on page 831](#)
- [Traceroute MIB on page 835](#)
- [Utility MIB on page 837](#)
- [VLAN MIB on page 841](#)
- [VPLS MIB on page 849](#)
- [VPN Certificate Objects MIB on page 859](#)
- [VPN MIB on page 863](#)

CHAPTER 4

AAA Objects MIB

- Interpreting the Enterprise-Specific AAA Objects MIB on page 67

Interpreting the Enterprise-Specific AAA Objects MIB

- AAA Objects MIB on page 67
- AAA Objects MIB Textual Conventions on page 68
- jnxUserAAAStatTable on page 69
- jnxUserAAAServerName on page 70
- Access Authentication-Related Traps on page 70
- jnxUserAAAAccessPool on page 71

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/junos13.2/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—{jnxUserAAAGlobalStats 2}

The object ID for address pools is as follows:

- **jnxUserAAAAccessPool 1**—{**jnxUserAAAAccessPoolTable 1**}

For information about the enterprise-specific AAA Objects MIB, see the following topics

- [AAA Objects MIB Textual Conventions on page 68](#)
- [jnxUserAAASatTable on page 69](#)
- [jnxUserAAAServerName on page 70](#)
- [Access Authentication-Related Traps on page 70](#)
- [jnxUserAAAAccessPool on page 71](#)

AAA Objects MIB Textual Conventions

jnxAuthenticateType—Defines the method to authenticate a user.

Table 5: jnxAuthenticateType

Method	Syntax Integer
none —no authentication	(0)
radius —authentication through a RADIUS server	(1)
local —local authentication	(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 6: 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)
securid —accounting through an RSA SecurID authentication server	(4)
jsrc —accounting through JSRC	(5)

jnxAuthorizationType—Defines the method to authorize a user.

Table 7: 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 8: 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)

jnxUserAAASatTable

jnxUserAAASatTable, whose object ID is {jnxUserAAAAccessAuthStats 1}, exposes the user authentication statistics listed in [Table 9 on page 70](#).

Table 9: jnxUserAAAStatTable

Object	Object ID	Description
jnxUserAAAStatEntry	jnxUserAAAStatTable 1	Statistics entries collected for authentication. Sequence of parameters: <ul style="list-style-type: none"> 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.

jnxUserAAAServerName

jnxUserAAAServerName, whose object ID is {jnxUserAAAAccessAuthStats 1}, specifies the server name that identifies the authentication server.

Access Authentication-Related Traps

Table 10 on page 70 identifies access authentication traps.

Table 10: 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.

jnxUserAAAAccessPool

jnxUserAAAAccessPoolTable, whose object ID is { **jnxUserAAAAccessPoolTable1** }, provides the ability to track address resource usage off-chassis. **jnxUserAAAAccessPoolTable** objects are listed in [Table 11 on page 71](#).

Table 11: jnxUserAAAAccessPoolTable

Object	Object ID	Description
jnxUserAAAAccessPoolRoutingInstance	jnxUserAAAAccessPoolEntry 1	Routing instance of the address pool.
jnxUserAAAAccessPoolName	jnxUserAAAAccessPoolEntry 2	Address pool name.
jnxUserAAAAccessPoolLinkName	jnxUserAAAAccessPoolEntry 3	Address pool link name.
jnxUserAAAAccessPoolFamilyType	jnxUserAAAAccessPoolEntry 4	Family type of this pool.
jnxUserAAAAccessPoolInetNetwork	jnxUserAAAAccessPoolEntry 5	Match criteria for this pool. Network or Prefix.
jnxUserAAAAccessPoolInetPrefixLength	jnxUserAAAAccessPoolEntry 6	Prefix length for an IPv6 pool.
jnxUserAAAAccessPoolOutOfMemory	jnxUserAAAAccessPoolEntry 7	Number of times this pool has flagged an Out of Memory condition.
jnxUserAAAAccessPoolOutOfAddresses	jnxUserAAAAccessPoolEntry 8	Number of times this pool has flagged an Out of Address condition.
jnxUserAAAAccessPoolAddressTotal	jnxUserAAAAccessPoolEntry 9	Total number of addresses or prefixes in this pool.
jnxUserAAAAccessPoolAddressesInUse	jnxUserAAAAccessPoolEntry 10	Total number of addresses or prefixes given out from this pool.
jnxUserAAAAccessPoolAddressUsage	jnxUserAAAAccessPoolEntry 11	Percentage of addresses used in this pool or linked pool. If this pool is the head of a linked chain of pools, this number reflects the usage for the entire chain. If this pool is part of a linked chain of pools, but not the head of the chain, the value is not used.

Table 11: jnxUserAAAAccessPoolTable (continued)

Object	Object ID	Description
jnxUserAAAAccessPoolAddressUsageHigh	jnxUserAAAAccessPoolEntry 12	Configured high percentage threshold of addresses used in this pool or linked pool. An SNMP trap is generated when this threshold is exceeded. This trap is generated only for unlinked pools or pools that are the head of a linked chain of pools. If this pool is part of a linked chain of pools, but not the head of the chain, then no traps are generated.
jnxUserAAAAccessPoolAddressUsageAbate	jnxUserAAAAccessPoolEntry 13	Configured abate percentage threshold of addresses used in this pool or linked pool. An SNMP trap clear is generated when address use falls below this threshold percentage. This trap is generated only for unlinked pools or pools that are the head of a linked chain of pools. If this pool is part of a linked chain of pools, but not the head of the chain, then no traps are generated.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 5

Access Authentication Objects MIB

- Interpreting the Enterprise-Specific Access Authentication Objects MIB on page 73

Interpreting the Enterprise-Specific Access Authentication Objects MIB

- Access Authentication Objects MIB on page 73
- jnxJsFwAuthStats on page 73
- jnxJsAuthTrapVars on page 74
- jnxJsAuthNotifications on page 74

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/junos13.2/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 73
- jnxJsAuthTrapVars on page 74
- jnxJsAuthNotifications on page 74

jnxJsFwAuthStats

jnxJsFwAuthStats, whose object ID is **{jnxJsAuthObjects 1}**, exposes the firewall authentication statistics listed in [Table 12 on page 74](#).

Table 12: 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.

jnxJsAuthTrapVars

jnxJsAuthTrapVars identifies access authentication traps variables listed in [Table 13 on page 74](#).

Table 13: jnxJsAuthTrapVars

Object	Object ID	Description
jnxJsFwAuthUserName	jnxJsAuthTrapVars 1	Name of the user who is attempting to be authenticated or has been authenticated.
jnxJsFwAuthServiceDesc	jnxJsAuthTrapVars 2	Service or application name that the authentication is performed for: Telnet, FTP, HTTP.
jnxJsFwAuthReason	jnxJsAuthTrapVars 3	Reason for the trap being generated—authentication failure due to: timeout, invalid password, invalid username, and so on.
jnxJsFwAuthClientIpAddr	jnxJsAuthTrapVars 4	Authentication client's IP address.

jnxJsAuthNotifications

jnxJsAuthNotifications, whose object ID is **{jnxJsAuthMIB 0}**, identifies the user access authentication notifications listed in [Table 14 on page 75](#).

Table 14: jnxJsAuthNotifications

Object	Object ID	Description
jnxJsFwAuthFailure	jnxJsAuthNotifications 1	<p>A firewall user authentication status trap to signify whether a user using the pass-through firewall authentication mechanism has been rejected due to the reason specified in the trap.</p> <ul style="list-style-type: none">• 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	<p>Firewall user authentication maximum capacity has been exceeded.</p> <p>jnxJsFwAuthTotalUsers indicates the total number of users being authenticated, and it has exceeded the maximum allowable users.</p>

- Related Documentation
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 6

Alarm MIB

- Interpreting the Enterprise-Specific Alarm MIB on page 77

Interpreting the Enterprise-Specific Alarm MIB

- Alarm MIB on page 77
- jnxAlarmRelayMode on page 77
- jnxYellowAlarms on page 78
- jnxRedAlarms on page 78

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/junos13.2/topics/reference/mibs/mib-jnx-chassis-alarm.txt.

For information about the enterprise-specific Alarm MIB objects, see the following topics:

- jnxAlarmRelayMode on page 77
- jnxYellowAlarms on page 78
- jnxRedAlarms on page 78

jnxAlarmRelayMode

The **jnxAlarmRelayMode**, whose object identifier is **{jnxCraftAlarms 1}**, denotes the alarm relay mode of the craft interface panel for the yellow and red alarms. This object uses the following integer values:

- **1—Other**: Other or unknown state
- **2—passOn**: Alarms are passed on. The alarm relay is activated to pass on the yellow or red alarms to audible sirens or visual flashing devices.
- **3—cutOff**: Alarms are turned off. Both the yellow and red alarms are cut off from the alarm relays and do not get passed on to audible sirens or visual flashing devices.



NOTE: Both the yellow and red alarms can be cut off from the alarm relay using a Alarm Cutoff/Lamp Test button on the front panel of the router chassis.

jnxYellowAlarms

The **jnxYellowAlarms**, whose object identifier is **{jnxCraftAlarms 2}**, contains the objects listed in [Table 15 on page 78](#).

Table 15: 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: <ul style="list-style-type: none"> other-1: The alarm state is unknown. off-2: The yellow alarms are off. You can turn off the yellow alarms using the Alarm Cutoff/Lamp Test button on the craft interface panel of the router chassis. 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 since the sysUp time was reset last time, or if the value is unknown.

jnxRedAlarms

The **jnxRedAlarms**, whose object identifier is **{jnxCraftAlarms 3}**, contains the objects listed in [Table 16 on page 78](#).

Table 16: 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: <ul style="list-style-type: none"> 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.

Table 16: jnxRedAlarms (*continued*)

Object	Object ID	Description
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**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 7

Analyzer MIB

- [Interpreting the Enterprise-Specific Analyzer MIB on page 81](#)

Interpreting the Enterprise-Specific Analyzer MIB

- [Analyzer MIB on page 81](#)
- [jnxAnalyzerTable on page 81](#)
- [jnxAnalyzerInputTable on page 82](#)
- [jnxAnalyzerOutputTable on page 83](#)

Analyzer MIB

The Juniper Networks enterprise-specific Analyzer MIB, whose object identifier is **{jnxExAnalyzer 1}**, contains analyzer and remote analyzer data related to port mirroring on the EX Series Ethernet Switches. Port mirroring is a method used on enterprise switches to monitor and analyze traffic on the network.

When port mirroring is enabled, copies of all (or a sample set of) packets are forwarded from one port of the switch to another port on the same switch (analyzer) or on another switch (remote analyzer) where the packet can be analyzed and studied.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-analyzer.txt.

For information about the enterprise-specific Analyzer MIB objects, see the following topics:

- [jnxAnalyzerTable on page 81](#)
- [jnxAnalyzerInputTable on page 82](#)
- [jnxAnalyzerOutputTable on page 83](#)

jnxAnalyzerTable

The **jnxAnalyzerTable**, whose object identifier is **{jnxAnalyzerMIBObjects 1}**, contains information about analyzer parameters. Each **jnxAnalyzerEntry** contains the objects listed in [Table 17 on page 82](#).

Table 17: jnxAnalyzerTable

Object	Object ID	Description
jnxAnalyzerName	jnxAnalyzerEntry 1	Uniquely identifies an analyzer configured on the switch.
jnxMirroringRatio	jnxAnalyzerEntry 2	Specifies the mirroring ratio. This object defines the sample size for mirroring. For example, 1 out of every <i>x</i> packets, where <i>x</i> is a number in the range of 1 through 2047.
jnxLossPriority	jnxAnalyzerEntry 3	Specifies the loss priority for a packet. When the number of packets at the analyzer port exceeds the bandwidth of the analyzer port, packets are dropped based on the loss priority value. When there is a bandwidth crunch, packets with high loss priority are dropped to clear the congestion. This object uses the following integer values to denote the loss priority: <ul style="list-style-type: none"> 0—Low loss priority 1—High loss priority

jnxAnalyzerInputTable

The **jnxAnalyzerInputTable** (obsolete in this release), whose object identifier is **{jnxAnalyzerMIBObjects 2}**, contains information about analyzer sessions. In a typical analyzer session, several source ports can be associated with a single destination port, and a range or series of ports can be mirrored.

Each **jnxAnalyzerInputEntry** provides information about input source ports, and contains the objects listed in [Table 18 on page 82](#).

Table 18: jnxAnalyzerInputTable

Object	Object ID	Description
jnxAnalyzerInputValue	jnxAnalyzerInputEntry 1	Identifies an analyzer input source port. This object can contain a display string of not more than 255 characters. <ul style="list-style-type: none"> If the value of jnxAnalyzerInputType is 1, then the value of jnxAnalyzerInputValue denotes the interface name of the input source. If the value of jnxAnalyzerInputType is 2, then the value of jnxAnalyzerInputValue denotes the VLAN name of the input source.
jnxAnalyzerInputOption	jnxAnalyzerInputEntry 2	Denotes the type of traffic to be mirrored from the source port; that is, whether it is ingress traffic or egress traffic. This object uses the following integer values: <ul style="list-style-type: none"> 1—Ingress traffic, where the analyzer monitors packets received by the source port. 2—Egress traffic, where the analyzer monitors packets transmitted by the source port. <p>In both the cases, the number of packets mirrored to the destination port depends on the jnxMirroringRatio.</p>

Table 18: jnxAnalyzerInputTable (*continued*)

Object	Object ID	Description
jnxAnalyzerInputType	jnxAnalyzerInputEntry 3	Denotes whether the mirroring source is an interface or a VLAN. This object uses integer values 1 (for interface) and 2 (for VLAN). For interfaces, you can configure either ingress or egress mirroring, whereas for VLANs, you can configure only ingress mirroring.

jnxAnalyzerOutputTable

The **jnxAnalyzerOutputTable**, whose object identifier is **{jnxAnalyzerMIBObjects 3}**, contains information about the destination port to which the packets are mirrored. Each **jnxAnalyzerOutputEntry** contains the objects listed in [Table 19 on page 83](#), and provides information about the destination port or destination VLAN.

Table 19: jnxAnalyzerOutputTable

Object	Object ID	Description
jnxAnalyzerOutputValue	jnxAnalyzerOutputEntry 1	Uniquely identifies a destination port or VLAN. This object can contain a string of not more than 255 characters. If the value of jnxAnalyzerOutputType is 1, then jnxAnalyzerOutputValue contains an interface name. If the value of jnxAnalyzerOutputType is 2, then jnxAnalyzerOutputValue contains a VLAN name.
jnxAnalyzerOutputType	jnxAnalyzerOutputEntry 2	Denotes the type of the output destination port. This object uses integer values 1 (for a destination port that is on the same switch) and 2 (for remote analyzer, that is a dedicated VLAN on a different switch).

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 8

Antivirus Objects MIB

- [Interpreting the Enterprise-Specific Antivirus Objects MIB on page 85](#)

Interpreting the Enterprise-Specific Antivirus Objects MIB

- [Antivirus Objects MIB on page 85](#)
- [Scan Engine Objects Table on page 86](#)
- [Scan Statistics Objects Table on page 86](#)
- [Antivirus Traps Objects Table on page 87](#)

Antivirus Objects MIB

The Juniper Networks enterprise-specific Antivirus Objects MIB, **jnxJsUtmAvMIB**, whose object ID is **{jnxJsUTMRoot 1}**, defines the MIB for antivirus monitoring.

The Antivirus Objects MIB is part of unified threat management (UTM).

Related MIB objects include the following:

- **jnxJsAntiVirusObjects**—**{jnxJsAntiVirus 1}**
- **jnxJsAntiVirusTraps**—**{jnxJsAntiVirus 2}**
- **jnxJsAntiVirusTrapVars**—**{jnxJsAntiVirus 3}**

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-utm-av.txt.

For information about the enterprise-specific Antivirus Objects MIB, see the following topics:

- [Scan Engine Objects Table on page 86](#)
- [Scan Statistics Objects Table on page 86](#)
- [Antivirus Traps Objects Table on page 87](#)

Scan Engine Objects Table

The **jnxJsAntiVirusEngine** object, whose object ID is **{jnxJsAntiVirusObjects 1}**, describes the objects in the Scan Engine Objects Table. See [Table 20 on page 86](#).

Table 20: Scan Engine Objects Table

Object	Object ID	Description
jnxJsAVEngineType	jnxJsAntiVirusEngine 1	The antivirus scan engine type in use. If antivirus is not configured, the engine type is not known. <ul style="list-style-type: none"> unknown-engine (1) kaspersky-lab-engine (2) juniper-express-engine (3)
jnxJsAVCurrentPatternVersionString	jnxJsAntiVirusEngine 2	The antivirus pattern database version currently in use.
jnxJsAVDatabaseType	jnxJsAntiVirusEngine 3	The database type in use: <ul style="list-style-type: none"> full (1) express (2) unknown (3)

Scan Statistics Objects Table

The **jnxJsAntiVirusStats** object, whose object ID is **{jnxJsAntiVirusObjects 2}**, describes the objects in the Scan Statistics Objects Table. See [Table 21 on page 86](#).

Table 21: Scan Statistics Objects Table

Object	Object ID	Description
jnxJsAVScanCodeClean	jnxJsAntiVirusStats 1	Number of clean requests passed by the antivirus scan engine.
jnxJsAVScanCodeInfected	jnxJsAntiVirusStats 2	Number of infected requests found by the antivirus scan engine.
jnxJsAVScanCodeProtected	jnxJsAntiVirusStats 3	Number of requests that the antivirus scan engine cannot scan because of password protection.
jnxJsAVScanCodeDecompress	jnxJsAntiVirusStats 4	Number of requests that the antivirus scan engine cannot scan because of those requests that exceeded the maximum decompression layer limit.
jnxJsAVScanCodeCorrupted	jnxJsAntiVirusStats 5	Number of requests that the antivirus scan engine cannot scan because of file corruption.

Table 21: Scan Statistics Objects Table (*continued*)

Object	Object ID	Description
<code>jnxJsAVScanCodeNoResource</code>	<code>jnxJsAntiVirusStats 6</code>	Number of requests that the antivirus scan engine cannot scan because of a lack of resources.
<code>jnxJsAVScanCodeInternalError</code>	<code>jnxJsAntiVirusStats 7</code>	Number of requests that the antivirus scan engine cannot scan because of an internal error.
<code>jnxJsAVScanCodeMaxContentSize</code>	<code>jnxJsAntiVirusStats 8</code>	Number of requests that the antivirus scan engine cannot scan because of those requests that exceeded the maximum content size limit.
<code>jnxJsAVScanCodeTooManyReq</code>	<code>jnxJsAntiVirusStats 9</code>	Number of requests that the antivirus scan engine cannot scan because of those requests that exceeded the maximum request limit.
<code>jnxJsAVScanCodeTimeout</code>	<code>jnxJsAntiVirusStats 10</code>	Number of requests that the antivirus scan engine cannot scan because of a scanning timeout.
<code>jnxJsAVScanCodeEngineNotReady</code>	<code>jnxJsAntiVirusStats 11</code>	Number of requests that the antivirus scan engine cannot scan because of the scan engine is not ready.

Antivirus Traps Objects Table

The `jnxJsAvPatternUpdateTrap` object, whose object ID is `{jnxJsAntiVirusTrapsPrefix 1}`, generates a trap when the antivirus pattern database is updated.

Table 22 on page 87 describes the objects in the Antivirus Traps Objects Table.

Table 22: Antivirus Traps Objects Table

Object	Object ID	Description
<code>jnxAVPatternVersionString</code>	<code>jnxJsAntiVirusTrapVars 1</code>	The last successful update of the antivirus pattern database version.
<code>jnxAVPatternTimestamp</code>	<code>jnxJsAntiVirusTrapVars 2</code>	The last successful update of the antivirus pattern database timestamp.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)

- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 9

ATM Class-of-Service MIB

- [Interpreting the Enterprise-Specific ATM Class-of-Service MIB on page 89](#)

Interpreting the Enterprise-Specific ATM Class-of-Service MIB

- [ATM Class-of-Service MIB on page 89](#)
- [jnxCosAtmVcTable on page 90](#)
- [jnxCosAtmVcScTable on page 90](#)
- [jnxCosAtmVcQstatsTable on page 92](#)
- [jnxCosAtmTrunkTable on page 93](#)

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 **jnxCoSFclF** (*Juniper Networks enterprise-specific CoS MIB*)

For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos13.2/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 90](#)
- [jnxCosAtmVcScTable on page 90](#)
- [jnxCosAtmVcQstatsTable on page 92](#)
- [jnxCosAtmTrunkTable on page 93](#)

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.

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 23 on page 90](#) lists the ATM scheduler parameters as represented by a **jnxCosAtmVcScEntry**.

Table 23: 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.

Table 23: jnxCosAtmVcScTable (continued)

Object	Object ID	Description
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) .
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) .

Table 23: jnxCosAtmVcScTable (continued)

Object	Object ID	Description
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) .

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 24 on page 92](#).

Table 24: 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.
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.

Table 24: jnxCosAtmVcQstatsTable (continued)

Object	Object ID	Description
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.

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 25 on page 93](#).

Table 25: jnxCosAtmTrunkTable

Object	Object ID	Description
jnxCosAtmTrunkMode	jnxCosAtmTrunkEntry 1	Represents the mode of CoS queue priority for the trunk: <ul style="list-style-type: none"> 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 .

Table 25: jnxCosAtmTrunkTable (continued)

Object	Object ID	Description
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.
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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 10

ATM MIB

- Interpreting the Enterprise-Specific ATM MIB on page 95

Interpreting the Enterprise-Specific ATM MIB

- ATM MIB on page 95
- jnxAtmIfTable on page 95
- jnxAtmVCTable on page 98
- jnxAtmVpTable on page 100
- jnxAtmTrunkTable on page 102

ATM MIB

The Juniper Networks enterprise-specific ATM MIB, whose object identifier is **{jnxMibs 10}**, extends the standard ATM MIB, RFC 1695, and contains information about ATM interfaces and VCs.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-atm.txt.

For information about the enterprise-specific ATM MIB objects, see the following topics:

- jnxAtmIfTable on page 95
- jnxAtmVCTable on page 98
- jnxAtmVpTable on page 100
- jnxAtmTrunkTable on page 102

jnxAtmIfTable

The **jnxAtmIfTable** augments the **atmInterfaceConfTable** in the standard ATM MIB. The object identifier for **jnxAtmIfTable** is **{jnxAtm 1}**. Each **jnxAtmIfEntry** in the **jnxAtmIfTable** contains the configuration and statistical information for a particular ATM port. The **jnxAtmIfEntry**, whose object identifier is **{jnxAtmIfTable 1}**, is an extension of the **atmInterfaceConfEntry** in the standard ATM MIB.

Table 26 on page 96 lists the objects associated with the **jnxAtmIfEntry**.

Table 26: jnxAtmIfTable

Object	Object ID	Description
jnxAtmIfPortType	jnxAtmIfEntry 1	<p>Represents the type of the physical port. This object uses the following integer values to denote the port type:</p> <ul style="list-style-type: none"> • 1 (other) • 2 (oc3) • 3 (oc1) • 4 (t3) • 5 (e3) • 6 (oc48)
jnxAtmIfEncaps	jnxAtmIfEntry 2	<p>Represents the type of ATM physical link layer encapsulation. This object uses the following integer values to denote the type of encapsulation:</p> <ul style="list-style-type: none"> • 1 (other) • 2 (atmPvc) • 3 (atmCccCellRelay)
jnxAtmIfLpBackInfo	jnxAtmIfEntry 3	<p>Represents the loopback configuration and type. This object uses the following integer values to denote the loopback configuration type:</p> <ul style="list-style-type: none"> • 1 (noLoopBack) • 2 (localLoopBack) • 3 (remoteLoopBack)
jnxAtmIfScrambleEnable	jnxAtmIfEntry 4	<p>Indicates whether scrambling is enabled (object value 1) or disabled (object value 2). Default value for this object is 2, disabled.</p>
jnxAtmIfTxCellCount	jnxAtmIfEntry 5	<p>Indicates the number of ATM cells, including the idle cells, transmitted by the interface.</p>
jnxAtmIfRxCellCount	jnxAtmIfEntry 6	<p>Indicates the number of ATM cells, excluding the idle cells, received by the interface.</p>
jnxAtmIfTxIdleCellCount	jnxAtmIfEntry 7	<p>Indicates the number of idle cells transmitted by the interface. When the interface does not have anything to send, it sends idle cells to fill the time slot.</p>
jnxAtmIfUncorrHCSErrs	jnxAtmIfEntry 8	<p>Indicates the number of uncorrectable cell Header Check Sequence (HCS) errors that occurred on the interface.</p>

Table 26: jnxAtmIfTable (*continued*)

Object	Object ID	Description
jnxAtmIfCorrHCSErrs	jnxAtmIfEntry 9	Indicates the number of correctable cell HCS errors.
jnxAtmIfTxCellFIFOOverRuns	jnxAtmIfEntry 10	Indicates the number of overruns in the Transmit FIFO.
jnxAtmIfRxCellFIFOOverRuns	jnxAtmIfEntry 11	Indicates the number of overruns in the Receive FIFO.
jnxAtmIfRxCellFIFOUnderRuns	jnxAtmIfEntry 12	Indicates the number of underruns in the Receive FIFO.
jnxAtmIfInInvalidVCCells	jnxAtmIfEntry 13	Indicates the number of cells that are arrived for a nonexistent VC.
jnxAtmIfInNoBufferOAMCells	jnxAtmIfEntry 14	Indicates the number of incoming OAM cells or raw cells that are dropped because of lack of buffer to handle them.
jnxAtmIfInNoBufDropPkts	jnxAtmIfEntry 15	Indicates the number of AAL5 packets that are dropped because of lack of buffer to handle them.
jnxAtmIfOutVCQueueDrops	jnxAtmIfEntry 16	Indicates the number of packets dropped because of queue limits on each VC.
jnxAtmIfInBadCrcs	jnxAtmIfEntry 17	Indicates the total number of incoming CRC errors.
jnxAtmIfInLenErrPkts	jnxAtmIfEntry 18	Indicates the number of AAL5 packets that were dropped because of incorrect length.
jnxAtmIfInTimeoutPkts	jnxAtmIfEntry 19	Indicates the number of AAL5 packets that were dropped because of reassembly timeout.
jnxAtmIfL2CircuitMode	jnxAtmIfEntry 20	Indicates the layer 2 circuit mode of the ATM interface (on an ATMII PIC). This object uses the following integer values to denote the circuit modes: <ul style="list-style-type: none"> • 1 (not applicable) • 2 (none) • 3 (aal5) • 4 (cell) • 5 (uniTrunk) • 6 (nniTrunk)

jnxAtmVCTable

The **jnxAtmVCTable**, whose object identifier is **{jnxAtm 2}**, extends the **atmVcTable** in the standard ATM MIB, and contains information about peer-to-peer, point-to-multipoint, and multicast virtual circuit entries.

Each **jnxAtmVCEntry**, whose object identifier is **{jnxAtmVCTable 1}**, in the **jnxAtmVCTable** contains the objects listed in [Table 27 on page 98](#).

However, the **jnxAtmVCEntry** does not store any statistic for multicast VCs. A value of zero indicates this. Similarly, a value of **0.0.0.0** indicates that the multipoint destination IP address for a P2P VC is invalid. If **atmCccCellRelay** is set as the encapsulation type for the logical interface (to which the VC belongs), only the generic VC info is valid. And, if OAM is disabled (indicated by zero value for **jnxAtmVCFlags**), then all the OAM config and f5 statistics information is invalid.

Table 27: jnxAtmVCTable

Object	Object ID	Description
jnxAtmVCTConnType	jnxAtmVCEntry 1	<p>Indicates the type of connection. This object uses the following integer values to denote the connection types:</p> <ul style="list-style-type: none"> 1 (other; unknown connection type or a connection type that is not one of the other connection types explicitly represented) 2 (peer-to-peer) 3 (point to multipoint) 4 (multicast)
jnxAtmVCEncapsulation	jnxAtmVCEntry 2	<p>Indicates the ATM encapsulation type associated with the VC. This object uses the following integer values to denote the encapsulation types:</p> <ul style="list-style-type: none"> 1 other 2 atmCccCellRelay (ATM cell relay for CCC) 3 atmCccVcMux (ATM VC for CCC) 4 atmCiscoNlpid (Cisco-compatible ATM NLPID encapsulation) 5 atmNlpid (ATM NLPID encapsulation) 6 atmSnap (ATM LLC/SNAP encapsulation) 7 atmVcMux (ATM VC multiplexing) 8 atmTccVcmux (Translational Cross Connection (TCC) over ATM VC MUX encapsulation) 9 atmTccSnap (TCC over ATM LLC/SNAP encapsulation)

Table 27: jnxAtmVCTable (continued)

Object	Object ID	Description
jnxAtmVCMpDestIPv4Addr	jnxAtmVCEntry 3	Represents the multipoint destination IPv4 address for a point-to-multipoint connection. This object returns an all zero address in the following cases: <ul style="list-style-type: none"> • When the connection type is not P2MP. • When the multipoint destination address type is IPv6.
jnxAtmVCMpDestIPv6Addr	jnxAtmVCEntry 4	Represents the multipoint destination IPv6 address for a P2MP connection. This object returns an all zero address in the following cases: <ul style="list-style-type: none"> • When the connection type is not P2MP. • When the multipoint destination address type is IPv4.
jnxAtmVCFlags	jnxAtmVCEntry 5	Contains the flags related to the VC.
jnxAtmVCTotalDownTime	jnxAtmVCEntry 6	Shows the total downtime for the VC after the last reboot of the system.
jnxAtmVCInBytes	jnxAtmVCEntry 7	Represents the number of bytes received on the VC.
jnxAtmVCOutBytes	jnxAtmVCEntry 8	Represents the number of bytes transmitted from the VC.
jnxAtmVCInPkts	jnxAtmVCEntry 9	Represents the number of packets received on the VC.
jnxAtmVCOutPkts	jnxAtmVCEntry 10	Represents the number of packets transmitted from the VC.
jnxAtmVCTailQueuePktDrops	jnxAtmVCEntry 11	Represents the number of packets that were dropped because of bandwidth constraints.
jnxAtmVCOAMPeriod	jnxAtmVCEntry 12	Shows the frequency at which the F5 cells are transmitted to check the status of the VC.
jnxAtmVCOAMUpCellCount	jnxAtmVCEntry 13	Shows the minimum number of loopback cells that are required to confirm that a VC is up.

Table 27: jnxAtmVCTable (continued)

Object	Object ID	Description
jnxAtmVCOAMDownCellCount	jnxAtmVCEntry 14	Shows the minimum number of loopback cells that are required to confirm that a VC is down. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCInOAMF5LoopCells	jnxAtmVCEntry 15	Shows the number of OAM F5 loopback cells received on a VC. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5LoopCells	jnxAtmVCEntry 16	Shows the number of OAM F5 loopback cells transmitted from a VC. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCInOAMF5RDICells	jnxAtmVCEntry 17	Shows the number of OAM F5 cells that are received with RDI (Remote Defect Indication) bit set. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5RDICells	jnxAtmVCEntry 18	Shows the number of OAM F5 cells that are transmitted with RDI (Remote Defect Indication) bit set. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCInOAMF5AISCells	jnxAtmVCEntry 19	Shows the number of OAM F5 cells that are received with alarm indication signal (AIS) bit set. NOTE: This object returns a zero value if OAM is not enabled.
jnxAtmVCOutOAMF5AISCells	jnxAtmVCEntry 20	Shows the number of OAM F5 cells that are transmitted with AIS bit set. NOTE: This object returns a zero value if OAM is not enabled.

jnxAtmVpTable

The **jnxAtmVpTable** extends the **atmVplTable** defined in RFC 2515, *Definitions of Managed Objects for ATM Management*, and contains additional information about ATM virtual paths (VP).

The **jnxAtmVpTable**, whose object identifier is **{jnxAtm 3}**, contains **jnxAtmVpEntry**. Each **jnxAtmVpEntry**, whose object ID is **{jnxAtmVpTable 1}**, contains the objects listed in [Table 28 on page 101](#).

Table 28: jnxAtmVpTable

Object	Object ID	Description
jnxAtmVpEntry	jnxAtmVpTable 1	<p>Represents configuration status and statistics information related to an ATM VP.</p> <p>However, traffic stats are available per VP tunnel only if shaping is configured on the VP. You can use the jnxAtmVpFlags to determine whether shaping is enabled.</p> <p>Similarly, the values for OAM config and OAM stat objects are invalid (default value: 0) if no OAM is configured. You can use the jnxAtmVpFlags to determine whether OAM is configured.</p> <p>NOTE: For an ATM-1 VP, the only valid object is jnxAtmVpFlags.</p>
jnxAtmVpFlags	jnxAtmVpEntry 1	<p>Represents the flags associated with the VP. This object uses the following values:</p> <ul style="list-style-type: none"> • 0 active • 1 down • 2 oamEnabled • 3 shapingEnabled • 4 passiveOam
jnxAtmVpTotalDownTime	jnxAtmVpEntry 2	Represents the total downtime for the VP since the last reboot of the system.
jnxAtmVpOamPeriod	jnxAtmVpEntry 3	<p>Indicates the frequency at which the OAM F4 cells are transmitted to find out the status of the VP.</p> <p>This object returns a value of zero if OAM is not enabled for the VP.</p>
jnxAtmVpOamUpCellCount	jnxAtmVpEntry 4	Indicates the minimum number of consecutive loopback cells required to confirm that a VP is up.
jnxAtmVpOamDownCellCount	jnxAtmVpEntry 5	Indicates the minimum number of consecutive loopback cells required to confirm that a VP is down.
jnxAtmVpInBytes	jnxAtmVpEntry 6	Indicates the number of bytes received on the VP.
jnxAtmVpOutBytes	jnxAtmVpEntry 7	Indicates the number of bytes sent out of the VP.
jnxAtmVpInPkts	jnxAtmVpEntry 8	Indicates the number of packets received on the VP.
jnxAtmVpOutPkts	jnxAtmVpEntry 9	Indicates the number of packets sent out on the VP.
jnxAtmVpInOamF4Cells	jnxAtmVpEntry 10	Indicates the number of OAM F4 cells received on the VP.

Table 28: jnxAtmVpTable (continued)

Object	Object ID	Description
jnxAtmVpOutOamF4Cells	jnxAtmVpEntry 11	Indicates the number of OAM F4 cells transmitted on the VP.
jnxAtmVpInOamF4LoopCells	jnxAtmVpEntry 12	Indicates the number of OAM F4 loopback cells received on the VP.
jnxAtmVpOutOamF4LoopCells	jnxAtmVpEntry 13	Indicates the number of OAM F4 cells transmitted on the VP.
jnxAtmVpInOamF4RdiCells	jnxAtmVpEntry 14	Indicates the number of OAM F4 RDI cells received on the VP.
jnxAtmVpOutOamF4RdiCells	jnxAtmVpEntry 15	Indicates the number of OAM F4 RDI cells transmitted on the VP.
jnxAtmVpInOamF4AisCells	jnxAtmVpEntry 16	Indicates the number of OAM F4 AIS cells received on the VP.

jnxAtmTrunkTable

The **jnxAtmTrunkTable**, whose object identifier is **{jnxAtm 4}**, contains information related to ATM trunks. Each **jnxAtmTrunkEntry** in **jnxAtmTrunkTable** contains the objects listed in [Table 29 on page 102](#).



NOTE: If the encapsulation type for the logical interface to which the trunk belongs is **atmCccCellRelay**, only the generic trunk information (**jnxAtmTrunkConnType**, **jnxAtmTrunkEncapsulation**, **nxAtmTrunkFlags**, and **jnxAtmTrunkTotalDownTime**) is valid.

Table 29: jnxAtmTrunkTable

Object	Object ID	Description
jnxAtmTrunkId	jnxAtmTrunkEntry 1	Represents the identifier of the ATM trunk.
jnxAtmTrunkConnType	jnxAtmTrunkEntry 2	Indicates the type of connection. This object uses the following integer values to denote the type of connection: <ul style="list-style-type: none"> 1 other 2 P2P
jnxAtmTrunkEncapsulation	jnxAtmTrunkEntry 3	Represents the ATM encapsulation type associated with the VC or trunk. This object uses the following integer values to denote the encapsulation type: <ul style="list-style-type: none"> 1 other 2 atmCccCellRelay
jnxAtmTrunkFlags	jnxAtmTrunkEntry 4	Represents the flags related to the trunk.
jnxAtmTrunkTotalDownTime	jnxAtmTrunkEntry 5	Indicates the total downtime (in seconds) for the trunk since the last reboot of the system.

Table 29: jnxAtmTrunkTable (*continued*)

Object	Object ID	Description
jnxAtmTrunkInBytes	jnxAtmTrunkEntry 6	Indicates the number of bytes received on the trunk.
jnxAtmTrunkOutBytes	jnxAtmTrunkEntry 7	Indicates the number of bytes sent out on the trunk.
jnxAtmTrunkInPkts	jnxAtmTrunkEntry 8	Indicates the number of packets received on the trunk.
jnxAtmTrunkOutPkts	jnxAtmTrunkEntry 9	Indicates the number of packets sent out on the trunk.
jnxAtmTrunkTailQueuePktDrops	jnxAtmTrunkEntry 10	Represents the number of packets that were dropped because of bandwidth constraints. This object indicates that the packets were queued to be transmitted at a rate faster than allowed.
jnxAtmTrunkInOAMF4AISCells	jnxAtmTrunkEntry 15	Indicates the number of OAM F4 cells that are received with AIS (Alarm Indication Signal) bit set.
jnxAtmTrunkOutOAMF4AISCells	jnxAtmTrunkEntry 16	Indicates the number of OAM F4 cells that are sent out with AIS bit set.

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 11

BGP4 V2 MIB

- Interpreting the Enterprise-Specific BGP4 V2 MIB on page 105

Interpreting the Enterprise-Specific BGP4 V2 MIB

- BGP4 V2 MIB on page 105
- jnxBgpM2PrefixCountersTable on page 105
- Enterprise-Specific BGP Traps on page 106

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/junos13.2/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 105
- Juniper Networks Enterprise-Specific BGP Traps on page 878

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 30 on page 106](#).

Table 30: 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.

Enterprise-Specific BGP Traps

For information about the enterprise-specific BGP traps, see “[Juniper Networks Enterprise-Specific BGP Traps](#)” on page 878.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 12

Bidirectional Forwarding Detection MIB

- Interpreting the Enterprise-Specific Bidirectional Forwarding Detection MIB on page 107

Interpreting the Enterprise-Specific Bidirectional Forwarding Detection MIB

- Bidirectional Forwarding Detection MIB on page 107
- jnxBfdSessTable on page 107
- Notifications for the BFD MIB on page 108

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/junos13.2/topics/reference/mibs/mib-jnx-bfd.txt.

For information about the enterprise-specific BFD MIB objects, see the following topics:

- jnxBfdSessTable on page 107
- Notifications for the BFD MIB on page 108

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 five objects, which are listed in [Table 31 on page 107](#).

Table 31: 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).

Table 31: jnxBfdSessTable (*continued*)

Object	Object Identifier	Description
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).
jnxBfdSessIntfName	jnxBfdSessEntry 5	The interface associated with the BFD session.

Notifications for the BFD MIB

Table 32 on page 108 lists the supported notifications for the BFD MIB.

Table 32: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 13

Chassis Cluster MIB

- [Interpreting the Enterprise-Specific Chassis Cluster MIB on page 109](#)

Interpreting the Enterprise-Specific Chassis Cluster MIB

- [Chassis Cluster MIB on page 109](#)
- [jnxJsChassisClusterSwitchover on page 109](#)
- [jnxJsChClusterIntfTrap on page 110](#)
- [Chassis Cluster Notifications on page 111](#)

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 *Junos OS System Basics and Services Command Reference*.

For information about the enterprise-specific chassis cluster MIB objects, see the following topics:

- [jnxJsChassisClusterSwitchover on page 109](#)
- [jnxJsChClusterIntfTrap on page 110](#)

jnxJsChassisClusterSwitchover

The **jnxJsChassisClusterSwitchover** trap, whose object identifier is **{jnxJsChassisClusterNotifications 1}**, contains notification information about signal switchover or failover.

[Table 33 on page 110](#) lists the objects associated with **jnxJsChassisClusterSwitchover**.

Table 33: Chassis Cluster Switchover Objects

Object	Object ID	Description
<code>jnxJsChClusterSwitchoverInfoRedundancyGroup</code>	<code>jnxJsChassisClusterTrapObjects</code> 1	Identification of the redundancy group that switched over.
<code>jnxJsChClusterSwitchoverInfoClusterId</code>	<code>jnxJsChassisClusterTrapObjects</code> 2	Identification number of the cluster.
<code>jnxJsChClusterSwitchoverInfoNodeId</code>	<code>jnxJsChassisClusterTrapObjects</code> 3	Node identification information where the switchover occurred.
<code>jnxJsChClusterSwitchoverInfoPreviousState</code>	<code>jnxJsChassisClusterTrapObjects</code> 4	Redundancy state of the cluster before the switchover.
<code>jnxJsChClusterSwitchoverInfoCurrentState</code>	<code>jnxJsChassisClusterTrapObjects</code> 5	Redundancy state of the cluster after the switchover.
<code>jnxJsChClusterSwitchoverInfoReason</code>	<code>jnxJsChassisClusterTrapObjects</code> 6	Reason for the switchover.

`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 34 on page 110 lists the objects associated with `jnxJsChClusterIntfTrap`.

Table 34: Chassis Cluster Trap Objects

Object	Object ID	Description
<code>jnxJsChClusterSwitchoverInfoClusterId</code>	<code>jnxJsChassisClusterTrapObjects</code> 2	Identification number of the cluster.
<code>jnxJsChClusterIntfName</code>	<code>jnxJsChassisClusterTrapObjects</code> 7	Name of the interface for which the state changed.
<code>jnxJsChClusterIntfState</code>	<code>jnxJsChassisClusterTrapObjects</code> 8	New state of the interface (up or down).
<code>jnxJsChClusterIntfSeverity</code>	<code>jnxJsChassisClusterTrapObjects</code> 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.
<code>jnxJsChClusterIntfStateReason</code>	<code>jnxJsChassisClusterTrapObjects</code> 10	Reason for the state change.

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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Chassis Forwarding MIB

- [Interpreting the Enterprise-Specific Chassis Forwarding MIB on page 113](#)

Interpreting the Enterprise-Specific Chassis Forwarding MIB

- [Chassis Forwarding MIB on page 113](#)
- [jnxFwddProcess on page 113](#)

Chassis Forwarding MIB

The Juniper Networks enterprise-specific Chassis Forwarding MIB, whose object identifier is **{jnxMibs 34}**, enables J Series Services Routers to fully support the Junos OS health monitor. This MIB extends the scope of health monitoring to include Junos OS forwarding process (**fwdd**) components on J Series Services Routers. The forwarding process is responsible for most of the packet transmission through a J Series Services Router. The overall performance of the router is largely determined by the effectiveness of the forwarding process.

The Junos OS health monitor uses objects in the Chassis Forwarding MIB to access information about the forwarding process such as microkernel CPU usage and real-time thread CPU usage.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-fwdd.txt.

For information about the enterprise-specific Chassis Forwarding MIB objects, see the following topic:

- [jnxFwddProcess on page 113](#)

jnxFwddProcess

The object identifier for **jnxFwddProcess** is **{jnxFwdd 1}**. This object shows health monitoring statistics for the forwarding process (**fwdd**). See [Table 35 on page 114](#).

Table 35: jnxFwddProcess

Object	Object Identifier	Description
jnxFwddMicroKernelCPUUsage	jnxFwddProcess 1	Percentage of the CPU being used by the forwarding process microkernel. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddRtThreadsCPUUsage	jnxFwddProcess 2	Percentage of the CPU being used by the forwarding process real-time threads. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddHeapUsage	jnxFwddProcess 3	Percentage of heap space being used by the forwarding process. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddDmaMemUsage	jnxFwddProcess 4	Percentage of DMA memory used by the forwarding process. If this information is unavailable or is not applicable, the value is 0 (zero).
jnxFwddUpTime	jnxFwddProcess 5	Forwarding process uptime expressed in terms of system uptime. If this information is unavailable or is not applicable, the value is 0 (zero).

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 15

Chassis MIB

- [Interpreting the Enterprise-Specific Chassis MIBs on page 115](#)

Interpreting the Enterprise-Specific Chassis MIBs

- [Chassis MIBs on page 116](#)
- [Chassis MIB Textual Conventions on page 117](#)
- [jnxBoxAnatomy on page 118](#)
- [Chassis Traps on page 346](#)
- [Chassis Definitions for Router Model MIB on page 352](#)
- [MIB Objects for the M120 Router on page 354](#)
- [MIB Objects for the MX2020 3D Universal Edge Router on page 356](#)
- [MIB Objects for the MX2010 3D Universal Edge Router on page 357](#)
- [MIB Objects for the MX960 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX480 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX240 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX80 3D Universal Edge Router on page 359](#)
- [MIB Objects for the MX40 3D Universal Edge Router on page 359](#)
- [MIB Objects for the MX10 3D Universal Edge Router on page 360](#)
- [MIB Objects for the MX5 3D Universal Edge Router on page 360](#)
- [MIB Objects for the PTX5000 Packet Transport Router on page 361](#)
- [MIB Objects for the QFX Series on page 362](#)
- [MIB Objects for the SRX1400 Services Gateway on page 364](#)
- [MIB Objects for the SRX3400 Services Gateway on page 365](#)
- [MIB Objects for the SRX3600 Services Gateway on page 365](#)
- [MIB Objects for the SRX5600 Services Gateway on page 366](#)
- [MIB Objects for the SRX5800 Services Gateway on page 366](#)
- [MIB Objects for the SRX100 Services Gateway on page 367](#)
- [MIB Objects for the SRX210 Services Gateway on page 367](#)
- [MIB Objects for the SRX220 Services Gateway on page 368](#)

- [MIB Objects for the SRX240 Services Gateway on page 368](#)
- [MIB Objects for the SRX550 Services Gateway on page 368](#)
- [MIB Objects for the SRX650 Services Gateway on page 369](#)

Chassis MIBs

The Juniper Networks enterprise-specific Chassis MIB provides information about 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/junos13.2/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 117](#)
- [jnxBoxAnatomy on page 118](#)
- [Chassis Traps on page 346](#)
- [Chassis Definitions for Router Model MIB on page 352](#)
- [MIB Objects for the M120 Router on page 354](#)
- [MIB Objects for the MX2010 3D Universal Edge Router on page 357](#)
- [MIB Objects for the MX2020 3D Universal Edge Router on page 356](#)
- [MIB Objects for the MX960 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX480 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX240 3D Universal Edge Router on page 358](#)
- [MIB Objects for the MX80 3D Universal Edge Router on page 359](#)
- [MIB Objects for the MX40 3D Universal Edge Router on page 359](#)
- [MIB Objects for the MX10 3D Universal Edge Router on page 360](#)
- [MIB Objects for the MX5 3D Universal Edge Router on page 360](#)
- *MIB Objects for the EX Series Ethernet Switches*
- [MIB Objects for the QFX Series on page 362](#)
- [MIB Objects for the SRX1400 Services Gateway on page 364](#)
- [MIB Objects for the SRX3400 Services Gateway on page 365](#)
- [MIB Objects for the SRX3600 Services Gateway on page 365](#)
- [MIB Objects for the SRX5600 Services Gateway on page 366](#)

- [MIB Objects for the SRX5800 Services Gateway on page 366](#)
- [MIB Objects for the SRX100 Services Gateway on page 367](#)
- [MIB Objects for the SRX210 Services Gateway on page 367](#)
- [MIB Objects for the SRX220 Services Gateway on page 368](#)
- [MIB Objects for the SRX240 Services Gateway on page 368](#)
- [MIB Objects for the SRX550 Services Gateway on page 368](#)
- [MIB Objects for the SRX650 Services Gateway on page 369](#)

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](#)
- [27—lcc14](#)
- [28—lcc15](#)

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 118](#)
- [jnxContainersTable on page 119](#)
- [jnxContentsLastChange on page 134](#)
- [jnxContentsTable on page 134](#)
- [jnxLEDLastChange on page 174](#)
- [jnxLEDTable on page 174](#)
- [jnxFilledLastChange on page 184](#)
- [jnxFilledTable on page 184](#)
- [jnxOperatingTable on page 215](#)
- [jnxRedundancyTable on page 250](#)
- [jnxFruTable on page 258](#)
- [jnxFruMib on page 345](#)
- [jnxBoxKernelMemoryUsedPercent on page 346](#)
- [jnxBoxSystemDomainType on page 346](#)

Top-Level Objects

The following branches of the **jnxBoxAnatomy** MIB are top-level objects:

- **jnxBoxClass**—The object identifier for the **jnxBoxClass** object is **{jnxBoxAnatomy 1}**. This object classifies the chassis product line.
- **jnxBoxDescr**—The object identifier for the **jnxBoxDescr** object is **{jnxBoxAnatomy 2}**. This object describes the chassis name and model.
- **jnxBoxSerialNo**—The object identifier for the **jnxBoxSerialNo** object is **{jnxBoxAnatomy 3}**. This object indicates the serial number of the chassis. **jnxBoxSerialNo** remains blank if the serial number is unknown or unavailable.

- **jnxBoxRevision**—The object identifier for the **jnxBoxRevision** object is **{jnxBoxAnatomy 4}**. This object indicates the last revision of the chassis.
- **jnxBoxInstalled**—The object identifier for the **jnxBoxInstalled** object is **{jnxBoxAnatomy 5}**. This object indicates the last time the box was installed and operational, represented by the **sysUpTime** value.

jnxContainersTable

The object identifier for the **jnxContainersTable** object is **{jnxBoxAnatomy 6}**. This object shows the structure of the chassis.

You can use the **jnxContainersTable** object to retrieve specific information 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 134](#).

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
- 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 39 on page 122](#) through [Table 52 on page 132](#) 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 121 describes the `jnxContainersEntry` objects contained in the `jnxContainersTable` of an M5 router.

Table 36: `jnxContainersEntry` Objects in the `jnxContainersTable` of an M5 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	<code>jnxChassisM5.0</code>	Chassis frame compartment	1
2	2	1	1	<code>jnxM5SlotPower.0</code>	Power supply compartment	2
4	3	1	1	<code>jnxM5SlotFan.0</code>	Fan compartment	4
6	2	1	1	<code>jnxM5SlotFEB.0</code>	FEB slot	1
7	1	1	1	<code>jnxM5SlotFPC.0</code>	FPC slot	1
8	1	2	7	<code>jnxM5MediaCardSlotPIC.0</code>	PIC space	4
9	2	1	1	<code>jnxM5SlotRE.0</code>	Routing Engine compartment	1

Table 37 on page 121 describes the `jnxContainersEntry` objects contained in the `jnxContainersTable` of an M10 router.

Table 37: `jnxContainersEntry` Objects in the `jnxContainersTable` of an M10 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	<code>jnxChassisM10.0</code>	Chassis frame compartment	1
2	2	1	1	<code>jnxM10SlotPower.0</code>	Power supply compartment	2
4	2	1	1	<code>jnxM10SlotFan.0</code>	Fan compartment	1
6	2	1	1	<code>jnxM10SlotFEB.0</code>	FEB slot	1
7	1	1	1	<code>jnxM10SlotFPC.0</code>	FPC slot	2
8	1	2	7	<code>jnxM10MediaCardSpacePIC.0</code>	PIC space	4
9	2	1	1	<code>jnxM10SlotRE.0</code>	Routing Engine compartment	1

Table 38 on page 122 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an M20 router.

Table 38: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM20.0	Chassis frame compartment	1
2	2	1	1	jnxM20SlotPower.0	Power supply compartment	2
4	3	1	1	jnxSlotFan.0	Fan compartment	4
6	2	1	1	jnxM20SlotSSB.0	SSB slot	2
7	1	1	1	jnxM20SlotFPC.0	FPC slot	4
8	1	2	7	jnxM20MediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM20RE.0	Routing Engine compartment	2
10	1	1	1	JNXM20FrontPanel.0	Front display slot	1

Table 39 on page 122 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an M40 router.

Table 39: 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

Table 39: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router (continued)

Index	View	Level	Within	Type	Description	Count
9	2	1	1	jnxSlotRoutingEngine.0	Routing Engine compartment	1

Table 40 on page 123 describes the jnxContainersEntry objects contained in the jnxContainersTable of an M160 router.

Table 40: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM160.0	Chassis frame compartment	1
2	2	1	1	Jnx160SlotPower.0	Power supply compartment	2
4	3	1	1	jnxM160SlotFan.0	Fan compartment	4
6	2	1	1	jnxM160SlotSFM.0	SFM slot	4
7	1	1	1	jnxM160SlotFPC.0	FPC slot	8
8	1	2	7	jnxM160MediaCardSlotPIC.0	PIC space	4
9	2	1	1	jnxM160SlotHM.0	Host slot	2
10	1	1	1	jnxM160SlotFPM.0	FPM slot	1
11	2	1	1	jnxM160SlotPCG.0	PCG slot	2
12	2	1	1	jnxM160SlotMCS.0	MCS slot	2
13	1	1	1	jnxM160SlotCIP.0	CIP slot	1

Table 41 on page 123 describes the jnxContainersEntry objects contained in the jnxContainersTable of an MX5 router.

Table 41: jnxContainersEntry Objects in the jnxContainersTable of an MX5 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisMX5.0	Chassis frame	1
2	2	1	1	jnxMX5SlotPower.0	PEM slot	2

Table 41: jnxContainersEntry Objects in the jnxContainersTable of an MX5 Router (continued)

Index	View	Level	Within	Type	Description	Count
4	3	1	1	jnxMX5SlotFan.0	Fan slot	1
6	1	1	1	jnxMX5SlotCFEB.0	TFEB slot	1
7	1	1	1	jnxMX5SlotFPC.0	FPC slot	2
8	1	2	7	jnxMX5MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX5SlotRE.0	Routing Engine slot	1
10	1	1	1	0.0		0
12	1	1	1	0.0	CB slot	0
20	1	2	7	jnxMX5MediaCardSpaceMIC.0	MIC Slot	2

Table 42 on page 124 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX10 router.

Table 42: jnxContainersEntry Objects in the jnxContainersTable of an MX10 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisMX10.0	Chassis frame	1
2	2	1	1	jnxMX10SlotPower.0	PEM slot	2
4	3	1	1	jnxMX10SlotFan.0	Fan slot	1
6	1	1	1	jnxMX10SlotCFEB.0	TFEB slot	1
7	1	1	1	jnxMX10SlotFPC.0	FPC slot	2
8	1	2	7	jnxMX10MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX10SlotRE.0	Routing Engine slot	1
10	1	1	1	0.0		0
12	1	1	1	0.0	CB slot	0
20	1	2	7	jnxMX10MediaCardSpaceMIC.0	MIC Slot	2

Table 43 on page 125 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX40 router.

Table 43: jnxContainersEntry Objects in the jnxContainersTable of an MX40 Router

Index	View	Level	With	Type	Description	Count
1	1	0	0	jnxChassisMX40.0	Chassis frame	1
2	2	1	1	jnxMX40SlotPower.0	PEM slot	2
4	3	1	1	jnxMX40SlotFan.0	Fan slot	1
6	1	1	1	jnxMX40SlotCFEB.0	TFEB slot	1
7	1	1	1	jnxMX40SlotFPC.0	FPC slot	2
8	1	2	7	jnxMX40MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX40SlotRE.0	Routing Engine slot	1
10	1	1	1	0.0		0
12	1	1	1	0.0	CB slot	0
20	1	2	7	jnxMX40MediaCardSpaceMIC.0	MIC Slot	2

Table 44 on page 125 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX80 router.

Table 44: jnxContainersEntry Objects in the jnxContainersTable of an MX80 Router

Index	View	Level	With	Type	Description	Count
1	1	0	0	jnxChassisMX80.0	Chassis frame	1
2	2	1	1	jnxMX80SlotPower.0	PEM slot	2
4	3	1	1	jnxMX80SlotFan.0	Fan slot	1
6	1	1	1	jnxMX80SlotCFEB.0	TFEB slot	1
7	1	1	1	jnxMX80SlotFPC.0	FPC slot	2
8	1	2	7	jnxMX80MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX80SlotRE.0	Routing Engine slot	1

Table 44: jnxContainersEntry Objects in the jnxContainersTable of an MX80 Router (continued)

Index	View	Level	With	Type	Description	Count
10	1	1	1	-	-	0
12	1	1	1	-	CB slot	0
20	1	2	7	jnxMX80MediaCardSpaceMIC0	MIC Slot	2

Table 45 on page 126 describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX240 router.

Table 45: jnxContainersEntry Objects in the jnxContainersTable of an MX240 Router

Index	View	Level	With	Type	Description	Count
1	1	0	0	jnxChassisMX240.0	Chassis frame	1
2	2	1	1	jnxMX240SlotPower.0	PEM slot	4
4	3	1	1	jnxMX240SlotFan.0	Fan slot	1
7	1	1	1	jnxMX240SlotFPC.0	FPC slot	3
8	1	2	7	jnxMX240MediaCardSpacePIC0	PIC slot	4
9	1	1	1	jnxMX240SlotHM.0	Routing Engine slot	2
10	1	1	1	jnxMX240SlotFPB.0	FPM slot	1
12	1	1	1	jnxMX240SlotCB.0	CB slot	2
20	1	2	7	jnxMX240MediaCardSpaceMIC0	MIC Slot	2

To verify the L1,L2 and L3 indexes, use the show chassis hardware command. Sample command output from an MX240 router is as follows:

```
user@host> show chassis hardware
```

Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			JN1120CBBAFC	MX240
Midplane	REV 07	760-021404	TR6747	MX240 Backplane
FPM Board	REV 03	760-021392	WX1884	Front Panel Display
PEM 0	Rev 03	740-017330	QCS08289016	PS 1.2-1.7kW; 100-240V
AC in				
Routing Engine 0	REV 07	740-015113	9009006883	RE-S-1300
Routing Engine 1	REV 07	740-015113	9009004502	RE-S-1300
CB 0	REV 03	710-021523	WV3731	MX SCB
CB 1	REV 03	710-021523	WV3653	MX SCB

FPC 2	REV 14	750-017679	KF5384	DPCE 20x 1GE R EQ
CPU	REV 03	710-022351	KG1955	DPC PMB
PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) EQ
Xcvr 0	REV 01	740-013111	8268748	SFP-T
Xcvr 1	REV 01	740-013111	8268827	SFP-T
Xcvr 2	REV 01	740-013111	8268860	SFP-T
Xcvr 3	REV 01	740-013111	8268847	SFP-T
Xcvr 4	REV 01	740-013111	8268776	SFP-T
Xcvr 5	REV 01	740-013111	8268838	SFP-T
Xcvr 6	REV 01	740-013111	8268765	SFP-T
Xcvr 7	REV 01	740-013111	8268770	SFP-T
Xcvr 8	REV 01	740-013111	8268768	SFP-T
Xcvr 9	REV 01	740-013111	8268814	SFP-T
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) EQ
Xcvr 0	REV 01	740-013111	8268886	SFP-T
Xcvr 1	REV 01	740-013111	8268771	SFP-T
Xcvr 2	REV 01	740-013111	8268856	SFP-T
Xcvr 3	REV 01	740-013111	8268782	SFP-T
Xcvr 4	REV 01	740-013111	8268945	SFP-T
Fan Tray 0	REV 01	710-021113	WW4083	MX240 Fan Tray

To verify the size of the memory, use the `show chassis fpc`, `show chassis routing-engine`, and `show chassis environment` commands. For more information about the output of these commands, see the *Junos OS Operational Mode Commands*.

```
user@host> show chassis fpc
```

Slot	State	Temp (C)	CPU Utilization (%)		Memory DRAM (MB)	Utilization (%)	
			Total	Interrupt		Heap	Buffer
0	Empty						
1	Empty						
2	Online	23	3	0	1024	12	20

```
user@host> show chassis routing-engine
```

Routing Engine status:

Slot 0:

Current state	Master
Election priority	Master (default)
Temperature	30 degrees C / 86 degrees F
CPU temperature	25 degrees C / 77 degrees F
DRAM	2048 MB (2048 MB installed)
Memory utilization	35 percent
CPU utilization:	
User	0 percent
Background	0 percent
Kernel	2 percent
Interrupt	0 percent
Idle	97 percent
Model	RE-S-1300
Serial ID	9009006883
Start time	2013-02-11 15:48:28 PST
Uptime	69 days, 6 hours, 51 minutes, 43 seconds
Last reboot reason	Router rebooted after a normal shutdown.
Load averages:	1 minute 5 minute 15 minute
	0.00 0.02 0.00

Routing Engine status:

Slot 1:

Current state	Backup
Election priority	Backup (default)
Temperature	32 degrees C / 89 degrees F
CPU temperature	25 degrees C / 77 degrees F
DRAM	2048

```

Memory utilization      25 percent
CPU utilization:
  User                  0 percent
  Background            0 percent
  Kernel                0 percent
  Interrupt             0 percent
  Idle                  99 percent
Model                  RE-S-1300
Serial ID              9009004502
Start time             2012-10-24 14:58:01 PDT
Uptime                 179 days, 8 hours, 49 minutes, 7 seconds
Last reboot reason     Router rebooted after a normal shutdown.
Load averages:         1 minute   5 minute  15 minute
                       0.00       0.00    0.00

```

```
user@host> show chassis environment
```

Class	Item	Status	Measurement
Temp	PEM 0	OK	30 degrees C / 86 degrees F
	PEM 1	Absent	
	PEM 2	Absent	
	PEM 3	Absent	
	Routing Engine 0	OK	30 degrees C / 86 degrees F
	Routing Engine 0 CPU	OK	25 degrees C / 77 degrees F
	Routing Engine 1	OK	32 degrees C / 89 degrees F
	Routing Engine 1 CPU	OK	25 degrees C / 77 degrees F
	CB 0 Intake	OK	25 degrees C / 77 degrees F
	CB 0 Exhaust A	OK	28 degrees C / 82 degrees F
	CB 0 Exhaust B	OK	29 degrees C / 84 degrees F
	CB 0 ACBC	OK	27 degrees C / 80 degrees F
	CB 0 SF A	OK	37 degrees C / 98 degrees F
	CB 0 SF B	OK	30 degrees C / 86 degrees F
	CB 1 Intake	OK	26 degrees C / 78 degrees F
	CB 1 Exhaust A	OK	27 degrees C / 80 degrees F
	CB 1 Exhaust B	OK	29 degrees C / 84 degrees F
	CB 1 ACBC	OK	27 degrees C / 80 degrees F
	CB 1 SF A	OK	39 degrees C / 102 degrees F
	CB 1 SF B	OK	31 degrees C / 87 degrees F
	FPC 2 Intake	OK	23 degrees C / 73 degrees F
	FPC 2 Exhaust A	OK	33 degrees C / 91 degrees F
	FPC 2 Exhaust B	OK	36 degrees C / 96 degrees F
	FPC 2 I3 0 TSensor	OK	38 degrees C / 100 degrees F
	FPC 2 I3 0 Chip	OK	43 degrees C / 109 degrees F
	FPC 2 I3 2 TSensor	OK	35 degrees C / 95 degrees F
	FPC 2 I3 2 Chip	OK	39 degrees C / 102 degrees F
	FPC 2 IA 0 TSensor	OK	36 degrees C / 96 degrees F
	FPC 2 IA 0 Chip	OK	30 degrees C / 86 degrees F
	FPC 2 IA 1 TSensor	OK	29 degrees C / 84 degrees F
	FPC 2 IA 1 Chip	OK	29 degrees C / 84 degrees F
Fans	Front Fan	OK	Spinning at normal speed
	Middle Fan	OK	Spinning at normal speed
	Rear Fan	OK	Spinning at normal speed

Table 46 on page 128 describes the `jnxContainersEntry` objects contained in the `jnxContainersTable` of an MX480 router.

Table 46: jnxContainersEntry Objects in the jnxContainersTable of an MX480 Router

Index	View	Level	With	Type	Description	Count
1	1	0	0	jnxChassisMX480.0	Chassis frame	1

Table 46: jnxContainersEntry Objects in the jnxContainersTable of an MX480 Router (continued)

Index	View	Level	Width	Type	Description	Count
2	2	1	1	jnxMX480SlotPower.0	PEM slot	4
4	3	1	1	jnxMX480SlotFan.0	Fan slot	1
7	1	1	1	jnxMX480SlotFPC.0	FPC slot	6
8	1	2	7	jnxMX480MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX480SlotHM.0	Routing Engine slot	2
10	1	1	1	jnxMX480SlotFPB.0	FPM slot	1
12	1	1	1	jnxMX480SlotCB.0	CB slot	2
20	1	2	7	jnxMX480MediaCardSpaceMIC.0	MIC Slot	2

[Table 47 on page 129](#) describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX960 router.

Table 47: jnxContainersEntry Objects in the jnxContainersTable of an MX960 Router

Index	View	Level	Width	Type	Description	Count
1	1	0	0	jnxChassisMX960.0	Chassis frame	1
2	2	1	1	jnxMX960SlotPower.0	PEM slot	5
4	3	1	1	jnxMX960SlotFan.0	Fan slot	2
7	1	1	1	jnxMX960SlotFPC.0	FPC slot	12
8	1	2	7	jnxMX960MediaCardSpacePIC.0	PIC slot	4
9	1	1	1	jnxMX960SlotHM.0	Routing Engine slot	2
10	1	1	1	jnxMX960SlotFPB.0	FPM slot	1
12	1	1	1	jnxMX960SlotCB.0	CB slot	3
20	1	2	7	jnxMX960MediaCardSpaceMIC.0	MIC Slot	2

[Table 48 on page 130](#) describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX2010 router.

Table 48: jnxContainersEntry Objects in the jnxContainersTable of an MX2010 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisMX2010.0	Chassis Frame	1
4	1	1	1	jnxMX2010SlotFan.0	FAN Slot	4
7	1	1	1	jnxMX2010SlotFPC.0	FPC Slot	10
8	1	2	7	jnxMX2010MediaCardSpacePIC.0	PIC Slot	4
9	1	1	1	jnxMX2010SlotHM.0	Routing Engine Slot	2
10	1	1	1	jnxMX2010SlotFPB.0	FPM Slot	1
12	1	1	1	jnxMX2010SlotCB.0	CB Slot	2
14	1	1	12	jnxMX2010SlotSPMB	SPMB Slot	2
15	1	1	12	jnxMX2010SlotSFB	SFB Slot	8
20	1	2	7	jnxMX2010MediaCardSpaceMIC.0	MIC Slot	2
21	1	1	1	jnxMX2010SlotPDM.0	PDM Slot	2
22	1	1	1	jnxMX2010SlotPSM.0	PSM Slot	9
23	1	1	1	jnxMX2010SlotADC.0	ADC Slot	10

[Table 49 on page 130](#) describes the **jnxContainersEntry** objects contained in the **jnxContainersTable** of an MX2020 router.

Table 49: jnxContainersEntry Objects in the jnxContainersTable of an MX2020 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisMX2020.0	Chassis Frame	1
4	1	1	1	jnxMX2020SlotFan.0	FAN Slot	4
7	1	1	1	jnxMX2020SlotFPC.0	FPC Slot	20
8	1	2	7	jnxMX2020MediaCardSpacePIC.0	PIC Slot	4
9	1	1	1	jnxMX2020SlotHM.0	Routing Engine Slot	2
10	1	1	1	jnxMX2020SlotFPB.0	FPM Slot	1

Table 49: jnxContainersEntry Objects in the jnxContainersTable of an MX2020 Router (continued)

Index	View	Level	Within	Type	Description	Count
12	1	1	1	jnxMX2020SlotCB.0	CB Slot	2
14	1	1	12	jnxMX2020SlotSPMB	SPMB Slot	2
15	1	1	12	jnxMX2020SlotSFB	SFB Slot	8
20	1	2	7	jnxMX2020MediaCardSpaceMIC.0	MIC Slot	2
21	1	1	1	jnxMX2020SlotPDM.0	PDM Slot	4
22	1	1	1	jnxMX2020SlotPSM.0	PSM Slot	18
23	1	1	1	jnxMX2020SlotADC.0	ADC Slot	20

Table 50 on page 131 describes the jnxContainersEntry objects contained in the jnxContainersTable of a T640 router.

Table 50: 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 51 on page 132](#) describes the `jnxContainersEntry` objects contained in the `jnxContainersTable` of a T320 router.

Table 51: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	<code>jnxChassisT320.0</code>	Chassis frame	1
2	2	1	1	<code>jnxT320SlotPower.0</code>	PEM slot	2
4	3	1	1	<code>jnxT320SlotFan.0</code>	Fan slot	3
7	1	1	1	<code>jnxT320SlotFPC.0</code>	FPC slot	8
8	1	2	7	<code>jnxT320MediaCardSpacePIC.0</code>	PIC slot	2
9	2	1	1	<code>jnxT320SlotHM.0</code>	Host slot	2
10	1	1	1	<code>jnxT320SlotFPB.0</code>	FPM slot	1
11	2	1	1	<code>jnxT320SlotSCG.0</code>	SCG slot	2
12	2	1	1	<code>jnxT320SlotCB.0</code>	CB slot	2
13	1	1	1	<code>jnxT320SlotCIP.0</code>	CIP slot	1
14	2	1	1	<code>jnxT320SlotSPMB.0</code>	SPMB slot	2
15	2	1	1	<code>jnxT320SlotSIB.0</code>	SIB slot	3

[Table 52 on page 132](#) describes the `jnxContainersEntry` objects contained in the `jnxContainersTable` of an M40e router.

Table 52: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router

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

Table 52: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router (continued)

Index	View	Level	Within	Type	Description	Count
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 53 on page 133 describes the jnxContainersEntry objects contained in the jnxContainersTable of a T4000 router.

Table 53: 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.



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**—Specifies 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/productshowroom/showroom/equip_id/carriers/overview.html. The object identifier is **{jnxContentsEntry 13}**.

Table 54 on page 135 through Table 58 on page 166 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 54 on page 135 provides an example of **jnxContentsEntry** objects contained in the **jnxContentTable** of an M20 router.

Table 54: 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

Table 54: 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
2	1	0	0	jnxM20PowerDC.0	DC power supply A	001652	REV 05	0:0:00:00.00	740-00146
2	2	0	0	jnxM20PowerDC.0	DC power supply B	001652	REV 05	0:0:00:00.00	740-00146
4	1	0	0	jnxM20Fan.0	Front top fan	—	—	0:0:00:00.00	—
4	2	0	0	jnxM20Fan	Middle fan	—	—	0:0:00:00.00	—
4	3	0	0	jnxM20Fan	Bottom fan	—	—	0:0:00:00.00	—
4	4	4	0	jnxM20Fan	Rear fan	—	—	0:0:00:00.00	—
6	1	0	0	jnxM20SSB.0	SSB 0 Internet Processor II	AG0809	REV 01	0:0:00:35.17	710-001951
7	1	0	0	jnxM20FPC.0	FPC @ 0/*/*	AN1335	REV 01	0:0:01:01.80	710-001292
7	2	0	0	jnxM20FPC.0	FPC @ 1/*/*	AN1124	REV 01	0:0:01:07.96	710-001292
7	3	0	0	jnxM20FPC.0	FPC @ 2/*/*	AN1726	REV 01	0:0:01:14:12	710-001292
7	4	0	0	jnxM20FPC.0	FPC @ 3/*/*	AN1691	REV 01	0:0:01:20.28	710-001292
8	1	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/0/*	HD4313	REV 04	0:0:00:00.00	750-002992
8	1	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/1/*	AJ5844	REV 04	0:0:00:00.00	750-002992
8	1	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/2/*	HD4518	REV 04	0:0:00:00.00	750-002992
8	1	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/3/*	HD4515	REV 04	0:0:00:00.00	750-002992

Table 54: 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	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
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	jnxM20ChStm11.0	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 C0C12, SMIR
PIC 3        REV 04   750-003250   AD9599      1x CSTM1, SMIR
```

Table 55 on page 138 provides an example of **jnxContentEntry** objects in the **jnxContentTable** of an MX2010 router.

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxBackplaneMX2010.0	Midplane	ABAB8506	REV01	0:0:00:00:00	750-044636
4	1	1	0	jnxMX2010Fan.0	Fan Tray 0 Fan 1	ACAY0015	REV 2A	0:0:00:00:00	760-046960
4	1	2	0	jnxMX2010Fan.0	Fan Tray 0 Fan 2	ACAY0015	REV 2A	0:0:00:00:00	760-046960
4	1	3	0	jnxMX2010Fan.0	Fan Tray 0 Fan 3	ACAY0015	REV 2A	0:0:00:00:00	760-046960
4	1	4	0	jnxMX2010Fan.0	Fan Tray 0 Fan 4	ACAY0015	REV 2A	0:0:00:00:00	760-046960

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	1	5	0	jnxMX2010Fan.0	Fan Tray 0 Fan 5	ACAY0015	REV 2A	0:0:00:00.00	760-046960
4	1	6	0	jnxMX2010Fan.0	Fan Tray 0 Fan 6	ACAY0015	REV 2A	0:0:00:00.00	760-046960
4	2	1	0	jnxMX2010Fan.0	Fan Tray 1 Fan 1	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	2	2	0	jnxMX2010Fan.0	Fan Tray 1 Fan 2	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	2	3	0	jnxMX2010Fan.0	Fan Tray 1 Fan 3	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	2	4	0	jnxMX2010Fan.0	Fan Tray 1 Fan 4	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	2	5	0	jnxMX2010Fan.0	Fan Tray 1 Fan 5	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	2	6	0	jnxMX2010Fan.0	Fan Tray 1 Fan 6	ACAY0019	REV 2A	0:0:00:00.00	760-046960
4	3	1	0	jnxMX2010Fan.0	Fan Tray 2 Fan 1	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	3	2	0	jnxMX2010Fan.0	Fan Tray 2 Fan 2	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	3	3	0	jnxMX2010Fan.0	Fan Tray 2 Fan 3	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	3	4	0	jnxMX2010Fan.0	Fan Tray 2 Fan 4	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	3	5	0	jnxMX2010Fan.0	Fan Tray 2 Fan 5	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	3	6	0	jnxMX2010Fan.0	Fan Tray 2 Fan 6	ACAY0020	REV 2A	0:0:00:00.00	760-046960
4	4	1	0	jnxMX2010Fan.0	Fan Tray 3 Fan 1	ACAY0021	REV 2A	0:0:00:00.00	760-046960
4	4	2	0	jnxMX2010Fan.0	Fan Tray 3 Fan 2	ACAY0021	REV 2A	0:0:00:00.00	760-046960

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	4	3	0	jnxMX2010Fan.0	Fan Tray 3 Fan 3	ACAY0021	REV 2A	0:0:00:00.00	760-046960
4	4	4	0	jnxMX2010Fan.0	Fan Tray 3 Fan 4	ACAY0021	REV 2A	0:0:00:00.00	760-046960
4	4	5	0	jnxMX2010Fan.0	Fan Tray 3 Fan 5	ACAY0021	REV 2A	0:0:00:00.00	760-046960
4	4	6	0	jnxMX2010Fan.0	Fan Tray 3 Fan 6	ACAY0021	REV 2A	0:0:00:00.00	760-046960
7	1	0	0	jnxMX2010FPC.0	FPC: MPC 3D16x10GE @ 0/*/*	CAAB1919	REV 33	0:0:11:31.73	750-028467
7	2	0	0	jnxMX2010FPC.0	FPC: MPC Type 3 @ 1/*/*	ZG5027	REV 21	0:0:11:41.46	750-033205
7	9	0	0	jnxMX2010FPC.0	FPC: MPC Type 2 3D @ 8/*/*	ZT9746	REV 22	0:0:11:47.21	750-031089
7	10	0	0	jnxMX2010FPC.0	FPC: MPC 3D16x10GE EM @ 9/*/*	ZL3591	REV 11	0:0:11:56.76	750-036284
8	1	1	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 0/0/*	BUILTIN	—	0:0:13:13.95	BUILTIN
8	1	2	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 0/1/*	BUILTIN	—	0:0:13:14.11	BUILTIN
8	1	3	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 0/2/*	BUILTIN	—	0:0:13:14.27	BUILTIN
8	1	4	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 0/3/*	BUILTIN	—	0:0:13:14.42	BUILTIN

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	2	1	0	jnxPicHyp10X10GESFPF	PIC: 10X10GE SFP @ 1/0/*	BUILTIN	REV 03	0:0:12:81.55	BUILTIN
8	2	3	0	jnxPicHyp10X10GESFPF	PIC: 10X10GE SFP @ 1/2/*	BUILTIN	REV 03	0:0:12:90.46	BUILTIN
8	9	1	0	jnxPicMICH10XGESFP	PIC: 10x 1GE(LAN) SFP @ 8/0/*	BUILTIN	REV 26	0:0:12:48.39	BUILTIN
8	9	2	0	jnxPicMICH10XGESFP	PIC: 10x 1GE(LAN) SFP @ 8/1/*	BUILTIN	REV 26	0:0:12:52.54	BUILTIN
8	9	3	0	jnxPicMICH2X10GEXFP	PIC: 2x10GE XFP @ 8/2/*	BUILTIN	REV 26	0:0:12:53.93	BUILTIN
8	9	4	0	jnxPicMICH2X10GEXFP	PIC: 2x10GE XFP @ 8/3/*	BUILTIN	REV 26	0:0:12:54.34	BUILTIN
8	10	1	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 9/0/*	BUILTIN	—	0:0:13:50.05	BUILTIN
8	10	2	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 9/1/*	BUILTIN	—	0:0:13:50.22	BUILTIN
8	10	3	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 9/2/*	BUILTIN	—	0:0:13:50.38	BUILTIN
8	10	4	0	jnxPicMIC4X10GESFPPlus	PIC: 4x 10GE(LAN) SFP+ @ 9/3/*	BUILTIN	—	0:0:13:50.54	BUILTIN
9	1	0	0	jnxMX2010HM.0	Routing Engine 0	9009094134	REV 02	0:0:00:00.00	740-041821

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
9	2	0	0	jnxMX2010HM.0	Routing Engine 1		–	0:0:00:00.00	–
10	1	1	0	jnxMX2010FPB.0	FPM Board	ZX8744	REV 06	0:0:00:00.00	711-032349
12	1	0	0	jnxMX2010CB.0	CB 0	CAAB3491	REV 08	0:0:00:00.00	750-040257
12	2	0	0	jnxMX2010CB.0	CB 1	CAAB3489	REV 08	0:0:00:00.00	750-040257
14	1	0	0	jnxMX2010SPMB.0	SPMB 0	CAAA6135	REV 02	0:0:10:04.75	711-041855
14	2	0	0	jnxMX2010SPMB.0	SPMB 1	CAAA6137	REV 02	0:0:10:01.37	711-041855
15	1	0	0	jnxMX2010SFB.0	SFB 0	ZV1828	REV 06	0:0:00:00.00	711-032385
15	2	0	0	jnxMX2010SFB.0	SFB 1	ZZ2568	REV 07	0:0:00:00.00	711-032385
15	3	0	0	jnxMX2010SFB.0	SFB 2	ZZ2563	REV 07	0:0:00:00.00	711-032385
15	4	0	0	jnxMX2010SFB.0	SFB 3	ZZ2564	REV 07	0:0:00:00.00	711-032385
15	5	0	0	jnxMX2010SFB.0	SFB 4	ZZ2580	REV 07	0:0:00:00.00	711-032385
15	6	0	0	jnxMX2010SFB.0	SFB 5	ZZ2579	REV 07	0:0:00:00.00	711-032385
15	7	0	0	jnxMX2010SFB.0	SFB 6	CAAB4882	REV 07	0:0:00:00.00	711-032385
15	8	0	0	jnxMX2010SFB.0	SFB 7	CAAB4898	REV 07	0:0:00:00.00	711-032385
20	2	1	0	jnxPicHyp10X10GESFPP	MIC: 10X10GE SFPP @ 1/0/*	ZV6299	REV 03	0:0:12:81.67	750-033307
20	2	2	0	jnxPicHyp10X10GESFPP	MIC: 10X10GE SFPP @ 1/1/*	ZV6268	REV 03	0:0:12:90.49	750-033307
20	9	1	0	jnxPicMIC20XGESFP	MIC: 3D 20x1GE (LAN) SFP @ 8/0/*	ABBS1150	REV 26	0:0:12:52.57	750-028392
20	9	2	0	jnxPicMIC4X10GEXFP	MIC: 3D 4x10GE XFP @ 8/1/*	ABBR9582	REV 26	0:0:12:54.34	750-028387

Table 55: jnxContentsEntry Objects in the jnxContentsTable of an MX2010 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
21	1	0	0	jnxMX2010PDM.0	PDM 0	WA00008	REV 0B	0:0:00:00.00	740-038109
21	2	0	0	jnxMX2010PDM.0	PDM 1	WA00014	REV 0B	0:0:00:00.00	740-038109
22	5	0	0	jnxMX2010PSM.0	PSM 4	VK00254	REV 0C	0:0:00:00.00	740-033727
22	6	0	0	jnxMX2010PSM.0	PSM 5	VG00015	REV 0B	0:0:00:00.00	740-033727
22	7	0	0	jnxMX2010PSM.0	PSM 6	VH00097	REV 0B	0:0:00:00.00	740-033727
22	8	0	0	jnxMX2010PSM.0	PSM 7	VJ00151	REV 0C	0:0:00:00.00	740-033727
22	9	0	0	jnxMX2010PSM.0	PSM 8	VJ00149	REV 0C	0:0:00:00.00	740-033727
23	1	0	0	jnxMX2010ADC.0	ADC 0	CAAC2073	REV 05	0:0:00:00.00	750-043596
23	2	0	0	jnxMX2010ADC.0	ADC 1	ZV4117	REV 01	0:0:00:00.00	750-043596
23	9	0	0	jnxMX2010ADC.0	ADC 8	ZV4107	REV 01	0:0:00:00.00	750-043596
23	10	0	0	jnxMX2010ADC.0	ADC 9	ZW1555	REV 02	0:0:00:00.00	750-043596

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an MX2010 router is as follows:

```

user@host> show chassis hardware
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN11E3217AFK  MX2010
Midplane      REV 01   750-044636   ABAB8506      Lower Backplane
Midplane 1    REV 01   711-044557   ZY8296        Upper Backplane
PMP           REV 03   711-032426   ACAJ1388      Power Midplane
FPM Board     REV 06   711-032349   ZX8744        Front Panel Display
PSM 4         REV 0C   740-033727   VK00254       DC 52V Power Supply
Module
PSM 5         REV 0B   740-033727   VG00015       DC 52V Power Supply
Module
PSM 6         REV 0B   740-033727   VH00097       DC 52V Power Supply
Module
PSM 7         REV 0C   740-033727   VJ00151       DC 52V Power Supply
Module
PSM 8         REV 0C   740-033727   VJ00149       DC 52V Power Supply
Module
PDM 0         REV 0B   740-038109   WA00008       DC Power Dist Module
PDM 1         REV 0B   740-038109   WA00014       DC Power Dist Module
Routing Engine 0 REV 02   740-041821   9009094134    RE-S-1800x4
Routing Engine 1 REV 02   740-041821   9009094141    RE-S-1800x4
CB 0          REV 08   750-040257   CAAB3491      Control Board
CB 1          REV 08   750-040257   CAAB3489      Control Board

```

SPMB 0	REV 02	711-041855	CAAA6135	PMB Board
SPMB 1	REV 02	711-041855	CAAA6137	PMB Board
SFB 0	REV 06	711-032385	ZV1828	Switch Fabric Board
SFB 1	REV 07	711-032385	ZZ2568	Switch Fabric Board
SFB 2	REV 07	711-032385	ZZ2563	Switch Fabric Board
SFB 3	REV 07	711-032385	ZZ2564	Switch Fabric Board
SFB 4	REV 07	711-032385	ZZ2580	Switch Fabric Board
SFB 5	REV 07	711-032385	ZZ2579	Switch Fabric Board
SFB 6	REV 07	711-032385	CAAB4882	Switch Fabric Board
SFB 7	REV 07	711-032385	CAAB4898	Switch Fabric Board
FPC 0	REV 33	750-028467	CAAB1919	MPC 3D 16x 10GE
CPU	REV 11	711-029089	CAAB7174	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AMH02RE	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMH038C	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AMH0390	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AMG0SUA	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AMH0579	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMG0SGP	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AMH04SV	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AMH04X3	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AMH0135	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMH02NC	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AMH02XB	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AMH02PN	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AMH057Y	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMG0JHE	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AMH02HT	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AMH04V4	SFP+-10G-SR
FPC 1	REV 21	750-033205	ZG5027	MPC Type 3
CPU	REV 04	711-035209	YT4780	HMPC PMB 2G
MIC 0	REV 03	750-033307	ZV6299	10X10GE SFPP
PIC 0		BUILTIN	BUILTIN	10X10GE SFPP
Xcvr 0	REV 01	740-031980	083363A00410	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	083363A00334	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	113363A00125	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	083363A00953	SFP+-10G-SR
Xcvr 4	REV 01	740-031980	AHR013D	SFP+-10G-SR
Xcvr 5	REV 01	740-031980	AJ40JUR	SFP+-10G-SR
Xcvr 6	REV 01	740-031980	AJ40JKL	SFP+-10G-SR
Xcvr 7	REV 01	740-031980	AJ30ECK	SFP+-10G-SR
Xcvr 8	REV 01	740-021308	19T511100864	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	19T511100868	SFP+-10G-SR
MIC 1	REV 03	750-033307	ZV6268	10X10GE SFPP
PIC 2		BUILTIN	BUILTIN	10X10GE SFPP
Xcvr 0	REV 01	740-031980	AJCOJML	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AJ403PC	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AJ10N25	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AJ40JF4	SFP+-10G-SR
Xcvr 4	REV 01	740-031980	AJ40JSJ	SFP+-10G-SR
Xcvr 5	REV 01	740-031980	AJ403V7	SFP+-10G-SR
Xcvr 6	REV 01	740-031980	AJ40JN3	SFP+-10G-SR
Xcvr 7	REV 01	740-031980	AJ40JSU	SFP+-10G-SR
Xcvr 8	REV 01	740-021308	19T511100468	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	19T511101363	SFP+-10G-SR
FPC 8	REV 22	750-031089	ZT9746	MPC Type 2 3D
CPU	REV 06	711-030884	ZS1271	MPC PMB 2G
MIC 0	REV 26	750-028392	ABBS1150	3D 20x 1GE(LAN) SFP

PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 01	740-031851	PLG023C	SFP-SX
Xcvr 1	REV 01	740-031851	PLG09C6	SFP-SX
Xcvr 2	REV 02	740-011613	AM0950SF9L7	SFP-SX
Xcvr 3	REV 02	740-011613	AM1001SFN1H	SFP-SX
Xcvr 4	REV 02	740-011613	AM1001SFM9D	SFP-SX
Xcvr 5	REV 02	740-011613	AM1001SFLTJ	SFP-SX
Xcvr 6	REV 01	740-031851	AC1108S03L9	SFP-SX
Xcvr 7	REV 01	740-031851	AC1102S00NC	SFP-SX
Xcvr 8	REV 01	740-031851	AC1102S00MX	SFP-SX
Xcvr 9	REV 01	740-031851	AC1102S0085	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 01	740-031851	AC1102S00KU	SFP-SX
Xcvr 1	REV 01	740-031851	AC1102S00NG	SFP-SX
Xcvr 2	REV 01	740-031851	AC1102S00K3	SFP-SX
Xcvr 3	REV 01	740-031851	AC1102S008R	SFP-SX
Xcvr 4	REV 01	740-031851	AM1107SUFVJ	SFP-SX
Xcvr 5	REV 01	740-031851	AC1108S03LG	SFP-SX
MIC 1	REV 26	750-028387	ABBR9582	3D 4x 10GE XFP
PIC 2		BUILTIN	BUILTIN	2x 10GE XFP
Xcvr 0		NON-JNPR	T10A91703	XFP-10G-SR
Xcvr 1		NON-JNPR	T09L42604	XFP-10G-SR
PIC 3		BUILTIN	BUILTIN	2x 10GE XFP
FPC 9	REV 11	750-036284	ZL3591	MPC 3D 16x 10GE EM
CPU	REV 10	711-029089	ZL0513	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	1YT517101825	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	1YT517101821	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	1YT517101682	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	ALQ13R6	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	1YT517101828	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	1YT517101716	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	1YT517101732	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	ALP0TR1	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	1YT517101741	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	1YT517101829	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	1YT517101669	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	ALQ14E3	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	1YT517101826	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	1YT517101817	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	1YT517101735	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	ALQ159A	SFP+-10G-SR
ADC 0	REV 05	750-043596	CAAC2073	Adapter Card
ADC 1	REV 01	750-043596	ZV4117	Adapter Card
ADC 8	REV 01	750-043596	ZV4107	Adapter Card
ADC 9	REV 02	750-043596	ZW1555	Adapter Card
Fan Tray 0	REV 2A	760-046960	ACAY0015	172mm FanTray - 6 Fans
Fan Tray 1	REV 2A	760-046960	ACAY0019	172mm FanTray - 6 Fans
Fan Tray 2	REV 2A	760-046960	ACAY0020	172mm FanTray - 6 Fans
Fan Tray 3	REV 2A	760-046960	ACAY0021	172mm FanTray - 6 Fans

Table 56 on page 146 provides an example of `jnxContentEntry` objects in the `jnxContentTable` of an MX2020 router.

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxBackplaneMX2020.0	Midplane	ABAC7440	REV04	0:0:00:00.00	711-032387
4	1	1	0	jnxMX2020Fan.0	Fan Tray 0 Fan 1	RABC0011	REV 02	0:0:00:00.00	760-042349
4	1	2	0	jnxMX2020Fan.0	Fan Tray 0 Fan 2	RABC0011	REV 02	0:0:00:00.00	760-042349
4	1	3	0	jnxMX2020Fan.0	Fan Tray 0 Fan 3	RABC0011	REV 02	0:0:00:00.00	760-042349
4	1	4	0	jnxMX2020Fan.0	Fan Tray 0 Fan 4	RABC0011	REV 02	0:0:00:00.00	760-042349
4	1	5	0	jnxMX2020Fan.0	Fan Tray 0 Fan 5	RABC0011	REV 02	0:0:00:00.00	760-042349
4	1	6	0	jnxMX2020Fan.0	Fan Tray 0 Fan 6	RABC0011	REV 02	0:0:00:00.00	760-042349
4	2	1	0	jnxMX2020Fan.0	Fan Tray 1 Fan 1	RABC0006	REV 02	0:0:00:00.00	760-042349
4	2	2	0	jnxMX2020Fan.0	Fan Tray 1 Fan 2	RABC0006	REV 02	0:0:00:00.00	760-042349
4	2	3	0	jnxMX2020Fan.0	Fan Tray 1 Fan 3	RABC0006	REV 02	0:0:00:00.00	760-042349
4	2	4	0	jnxMX2020Fan.0	Fan Tray 1 Fan 4	RABC0006	REV 02	0:0:00:00.00	760-042349
4	2	5	0	jnxMX2020Fan.0	Fan Tray 1 Fan 5	RABC0006	REV 02	0:0:00:00.00	760-042349
4	2	6	0	jnxMX2020Fan.0	Fan Tray 1 Fan 6	RABC0006	REV 02	0:0:00:00.00	760-042349
4	3	1	0	jnxMX2020Fan.0	Fan Tray 2 Fan 1	RABC0010	REV 02	0:0:00:00.00	760-042349
4	3	2	0	jnxMX2020Fan.0	Fan Tray 2 Fan 2	RABC0010	REV 02	0:0:00:00.00	760-042349
4	3	3	0	jnxMX2020Fan.0	Fan Tray 2 Fan 3	RABC0010	REV 02	0:0:00:00.00	760-042349

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	3	4	0	jnxMX2020Fan.0	Fan Tray 2 Fan 4	RABC0010	REV 02	0:0:00:00.00	760-042349
4	3	5	0	jnxMX2020Fan.0	Fan Tray 2 Fan 5	RABC0010	REV 02	0:0:00:00.00	760-042349
4	3	6	0	jnxMX2020Fan.0	Fan Tray 2 Fan 6	RABC0010	REV 02	0:0:00:00.00	760-042349
4	4	1	0	jnxMX2020Fan.0	Fan Tray 3 Fan 1	RABC0003	REV 02	0:0:00:00.00	760-042349
4	4	2	0	jnxMX2020Fan.0	Fan Tray 3 Fan 2	RABC0003	REV 02	0:0:00:00.00	760-042349
4	4	3	0	jnxMX2020Fan.0	Fan Tray 3 Fan 3	RABC0003	REV 02	0:0:00:00.00	760-042349
4	4	4	0	jnxMX2020Fan.0	Fan Tray 3 Fan 4	RABC0003	REV 02	0:0:00:00.00	760-042349
4	4	5	0	jnxMX2020Fan.0	Fan Tray 3 Fan 5	RABC0003	REV 02	0:0:00:00.00	760-042349
4	4	6	0	jnxMX2020Fan.0	Fan Tray 3 Fan 6	RABC0003	REV 02	0:0:00:00.00	760-042349
7	1	0	0	jnxMX2020FPC.0	FPC: MPC Type 4-1 @ 0/*/*	CAAE2200	REV 11	0:0:01:88.91	750-037358
7	2	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 1/*/*	YE2633	REV 22	0:0:01:93.81	750-028467
7	3	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 2/*/*	ZR1986	REV 32	0:0:02:00.68	750-028467
7	4	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 3/*/*	ZR1942	REV 32	0:0:02:08.06	750-028467
7	5	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 4/*/*	ZR2032	REV 32	0:0:02:16.71	750-028467

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	6	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 5/*/*	ZR2012	REV 32	0:0:02:26.87	750-028467
7	7	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 6/*/*	ZR2121	REV 32	0:0:02:32.31	750-028467
7	8	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 7/*/*	ZS3655	REV 32	0:0:02:41.90	750-028467
7	9	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 8/*/*	ABBP2238	REV 32	0:0:02:51.39	750-028467
7	10	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 9/*/*	ZR2077	REV 32	0:0:02:55.84	750-028467
7	11	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 10/*/*	XX3381	REV 19	0:0:02:66.63	750-028467
7	12	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 11/*/*	ZR2137	REV 32	0:0:02:75.55	750-028467
7	13	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 12/*/*	ABBP1763	REV 32	0:0:02:84.26	750-028467
7	14	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 13/*/*	ABBP2197	REV 32	0:0:02:92.95	750-028467
7	15	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 14/*/*	YE2664	REV 22	0:0:03:02.72	750-028467
7	16	0	0	jnxMX2020FPC.0	FPC: MPC Type 3 @ 15/*/*	ZL5997	REV 27	0:0:03:11.96	750-033205
7	17	0	0	jnxMX2020FPC.0	FPC: MPC 3D 16x 10GE @ 16/*/*	ABBH8157	REV 26	0:0:03:18.01	750-028467

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
7	18	0	0	jnxMX2020FPC.0	FPC: MPC 3D16x10GE @ 17/*/*	ZR2002	REV 32	0:0:03:26:10	750-028467
7	19	0	0	jnxMX2020FPC.0	FPC: MPC Type 4-2 @ 18/*/*	CAAF0949	REV 09	0:0:03:36:93	750-037355
7	20	0	0	jnxMX2020FPC.0	FPC: MPC 3D16x10GE @ 19/*/*	ZR2079	REV 32	0:0:03:44:25	750-028467
8	1	1	0	jnxPic8x10GESFPPMIC	PIC: 8X10GE SFPP @ 0/0/*	BUILTIN	—	0:0:03:85:91	BUILTIN
8	1	2	0	jnxPic8x10GESFPPMIC	PIC: 8X10GE SFPP @ 0/1/*	BUILTIN	—	0:0:03:93:07	BUILTIN
8	1	3	0	jnxPic8x10GESFPPMIC	PIC: 8X10GE SFPP @ 0/2/*	BUILTIN	—	0:0:03:93:18	BUILTIN
8	1	4	0	jnxPic8x10GESFPPMIC	PIC: 8X10GE SFPP @ 0/3/*	BUILTIN	—	0:0:03:93:67	BUILTIN
8	2	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 1/0/*	BUILTIN	—	0:0:03:83:53	BUILTIN
8	2	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 1/1/*	BUILTIN	—	0:0:03:83:70	BUILTIN
8	2	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 1/2/*	BUILTIN	—	0:0:03:83:86	BUILTIN
8	2	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 1/3/*	BUILTIN	—	0:0:03:84:01	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	3	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 2/0/*	BUILTIN	—	0:0:03:83:93	BUILTIN
8	3	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 2/1/*	BUILTIN	—	0:0:03:84:09	BUILTIN
8	3	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 2/2/*	BUILTIN	—	0:0:03:84:25	BUILTIN
8	3	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 2/3/*	BUILTIN	—	0:0:03:84:41	BUILTIN
8	4	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 3/0/*	BUILTIN	—	0:0:03:89:93	BUILTIN
8	4	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 3/1/*	BUILTIN	—	0:0:03:90:10	BUILTIN
8	4	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 3/2/*	BUILTIN	—	0:0:03:90:26	BUILTIN
8	4	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 3/3/*	BUILTIN	—	0:0:03:90:42	BUILTIN
8	5	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 4/0/*	BUILTIN	—	0:0:03:98:60	BUILTIN
8	5	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 4/1/*	BUILTIN	—	0:0:03:98:76	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	5	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 4/2/*	BUILTIN	—	0:0:03:98:92	BUILTIN
8	5	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 4/3/*	BUILTIN	—	0:0:03:99:07	BUILTIN
8	6	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 5/0/*	BUILTIN	—	0:0:04:07:74	BUILTIN
8	6	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 5/1/*	BUILTIN	—	0:0:04:07:90	BUILTIN
8	6	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 5/2/*	BUILTIN	—	0:0:04:08:07	BUILTIN
8	6	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 5/3/*	BUILTIN	—	0:0:04:08:22	BUILTIN
8	7	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 6/0/*	BUILTIN	—	0:0:04:14:16	BUILTIN
8	7	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 6/1/*	BUILTIN	—	0:0:04:14:32	BUILTIN
8	7	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 6/2/*	BUILTIN	—	0:0:04:14:48	BUILTIN
8	7	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 6/3/*	BUILTIN	—	0:0:04:14:64	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	8	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 7/0/*	BUILTIN	—	0:0:04:23:74	BUILTIN
8	8	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 7/1/*	BUILTIN	—	0:0:04:23:91	BUILTIN
8	8	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 7/2/*	BUILTIN	—	0:0:04:24:07	BUILTIN
8	8	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 7/3/*	BUILTIN	—	0:0:04:24:23	BUILTIN
8	9	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 8/0/*	BUILTIN	—	0:0:04:39:47	BUILTIN
8	9	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 8/1/*	BUILTIN	—	0:0:04:39:63	BUILTIN
8	9	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 8/2/*	BUILTIN	—	0:0:04:39:79	BUILTIN
8	9	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 8/3/*	BUILTIN	—	0:0:04:39:95	BUILTIN
8	10	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 9/0/*	BUILTIN	—	0:0:04:39:93	BUILTIN
8	10	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 9/1/*	BUILTIN	—	0:0:04:40:08	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	10	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 9/2/*	BUILTIN	—	0:0:04:40:25	BUILTIN
8	10	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 9/3/*	BUILTIN	—	0:0:04:40:40	BUILTIN
8	11	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 10/0/*	BUILTIN	—	0:0:04:59:71	BUILTIN
8	11	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 10/1/*	BUILTIN	—	0:0:04:59:87	BUILTIN
8	11	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 10/2/*	BUILTIN	—	0:0:04:60:03	BUILTIN
8	11	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 10/3/*	BUILTIN	—	0:0:04:60:19	BUILTIN
8	12	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 11/0/*	BUILTIN	—	0:0:04:56:66	BUILTIN
8	12	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 11/1/*	BUILTIN	—	0:0:04:56:81	BUILTIN
8	12	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 11/2/*	BUILTIN	—	0:0:04:57:04	BUILTIN
8	12	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 11/3/*	BUILTIN	—	0:0:04:57:14	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	13	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 12/0/*	BUILTIN	—	0:0:04:67:33	BUILTIN
8	13	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 12/1/*	BUILTIN	—	0:0:04:67:48	BUILTIN
8	13	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 12/2/*	BUILTIN	—	0:0:04:67:65	BUILTIN
8	13	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 12/3/*	BUILTIN	—	0:0:04:67:80	BUILTIN
8	14	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 13/0/*	BUILTIN	—	0:0:04:84:05	BUILTIN
8	14	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 13/1/*	BUILTIN	—	0:0:04:84:21	BUILTIN
8	14	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 13/2/*	BUILTIN	—	0:0:04:84:37	BUILTIN
8	14	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 13/3/*	BUILTIN	—	0:0:04:84:52	BUILTIN
8	15	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 14/0/*	BUILTIN	—	0:0:04:93:80	BUILTIN
8	15	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 14/1/*	BUILTIN	—	0:0:04:93:96	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	15	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 14/2/*	BUILTIN	—	0:0:04:94:13	BUILTIN
8	15	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 14/3/*	BUILTIN	—	0:0:04:94:28	BUILTIN
8	16	1	0	jnxPicHyp1X100GECFP	PIC: 1X100GE CFP @ 15/0/*	BUILTIN	REV 05	0:0:04:54:90	BUILTIN
8	16	3	0	jnxPicHyp1X100GECFP	PIC: 1X100GE CFP @ 15/2/*	BUILTIN	REV 02	0:0:04:57:01	BUILTIN
8	17	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 16/0/*	BUILTIN	—	0:0:05:10:11	BUILTIN
8	17	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 16/1/*	BUILTIN	—	0:0:05:10:28	BUILTIN
8	17	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 16/2/*	BUILTIN	—	0:0:05:10:44	BUILTIN
8	17	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 16/3/*	BUILTIN	—	0:0:05:10:59	BUILTIN
8	18	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 17/0/*	BUILTIN	—	0:0:05:08:14	BUILTIN
8	18	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 17/1/*	BUILTIN	—	0:0:05:08:30	BUILTIN

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	18	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 17/2/*	BUILTIN	—	0:0:05:08:46	BUILTIN
8	18	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 17/3/*	BUILTIN	—	0:0:05:08:62	BUILTIN
8	19	1	0	jnxPic4x10GESFPPMIC	PIC: 4x10GE SFPP @ 18/0/*	BUILTIN	—	0:0:05:33:63	BUILTIN
8	19	2	0	jnxPic1x100GESFPPMIC	PIC: 1X100GE CFP @ 18/1/*	BUILTIN	—	0:0:05:40:80	BUILTIN
8	19	3	0	jnxPic4x10GESFPPMIC	PIC: 4x10GE SFPP @ 18/2/*	BUILTIN	—	0:0:05:41:00	BUILTIN
8	19	4	0	jnxPic1x100GECFPMIC	PIC: 1X100GE CFP @ 18/3/*	BUILTIN	—	0:0:05:41:46	BUILTIN
8	20	1	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 19/0/*	BUILTIN	—	0:0:05:25:37	BUILTIN
8	20	2	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 19/1/*	BUILTIN	—	0:0:05:25:52	BUILTIN
8	20	3	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 19/2/*	BUILTIN	—	0:0:05:25:69	BUILTIN
8	20	4	0	jnxPicMIC4X10GESFPPLUS	PIC: 4x 10GE(LAN) SFP+ @ 19/3/*	BUILTIN	—	0:0:05:25:84	BUILTIN
9	1	0	0	jnxMX2020HM.0	Routing Engine 0	9009089721	REV 02	0:0:00:00:00	740-041821

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
9	2	0	0	jnxMX2020HM.0	Routing Engine 1	9009089721	REV 02	0:0:00:00.00	740-041821
10	1	1	0	jnxMX2020FPB.0	FPM Board	ZM0068	REV 03	0:0:00:00.00	711-032349
12	1	0	0	jnxMX2020CB.0	CB 0	ZT2871	REV 04	0:0:00:00.00	750-040257
12	2	0	0	jnxMX2020CB.0	CB 1	ZT2854	REV 04	0:0:00:00.00	750-040257
14	1	0	0	jnxMX2020SPMB.0	SPMB 0	ZS2262	REV 01	0:0:00:18.84	711-041855
14	2	0	0	jnxMX2020SPMB.0	SPMB 1	ZS2265	REV 01	0:0:00:29.12	711-041855
15	1	0	0	jnxMX2020SFB.0	SFB 0	ZV4232	REV 04	0:0:00:00.00	711-032385
15	2	0	0	jnxMX2020SFB.0	SFB 1	ZV4213	REV 04	0:0:00:00.00	711-032385
15	3	0	0	jnxMX2020SFB.0	SFB 2	ZV4215	REV 04	0:0:00:00.00	711-032385
15	4	0	0	jnxMX2020SFB.0	SFB 3	ZB7638	REV 01	0:0:00:00.00	711-032385
15	5	0	0	jnxMX2020SFB.0	SFB 4	ZV4206	REV 04	0:0:00:00.00	711-032385
15	6	0	0	jnxMX2020SFB.0	SFB 5	ZV4204	REV 04	0:0:00:00.00	711-032385
15	7	0	0	jnxMX2020SFB.0	SFB 6	ZV4189	REV 04	0:0:00:00.00	711-032385
15	8	0	0	jnxMX2020SFB.0	SFB 7	ZV4207	REV 04	0:0:00:00.00	711-032385
20	16	1	0	jnxPicHyp1X100GECFP	MIC: 1X100GE CFP @ 15/0/*	YR3292	REV 05	0:0:04:54.91	740-038109
20	16	2	0	jnxPicHyp1X100GECFP	MIC: 1X100GE CFP @ 15/1/*	YG3239	REV 02	0:0:04:57.04	740-038109
21	1	0	0	jnxMX2020PDM.0	PDM 0	VJ00013	REV 0B	0:0:00:00.00	740-038109
21	4	0	0	jnxMX2020PDM.0	PDM 3	VJ00004	REV 0B	0:0:00:00.00	740-038109
22	3	0	0	jnxMX2020PSM.0	PSM 2	VK00212	REV 0C	0:0:00:00.00	740-033727
22	4	0	0	jnxMX2020PSM.0	PSM 3	VK00186	REV 0C	0:0:00:00.00	740-033727

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
22	5	0	0	jnxMX2020PSM.0	PSM 4	VK00187	REV 0C	0:0:00:00.00	740-033727
22	6	0	0	jnxMX2020PSM.0	PSM 5	VK00292	REV 0C	0:0:00:00.00	740-033727
22	7	0	0	jnxMX2020PSM.0	PSM 6	VK00190	REV 0C	0:0:00:00.00	740-033727
22	8	0	0	jnxMX2020PSM.0	PSM 7	VK00189	REV 0C	0:0:00:00.00	740-033727
22	9	0	0	jnxMX2020PSM.0	PSM 8	1E012130013	REV 01	0:0:00:00.00	740-033727
22	12	0	0	jnxMX2020PSM.0	PSM 11	VK00211	REV 0C	0:0:00:00.00	740-033727
22	13	0	0	jnxMX2020PSM.0	PSM 12	VK00316	REV 0C	0:0:00:00.00	740-033727
22	14	0	0	jnxMX2020PSM.0	PSM 13	VK00184	REV 0C	0:0:00:00.00	740-033727
22	15	0	0	jnxMX2020PSM.0	PSM 14	VK00310	REV 0C	0:0:00:00.00	740-033727
22	16	0	0	jnxMX2020PSM.0	PSM 15	VK00291	REV 0C	0:0:00:00.00	740-033727
22	17	0	0	jnxMX2020PSM.0	PSM 16	VK00185	REV 0C	0:0:00:00.00	740-033727
22	18	0	0	jnxMX2020PSM.0	PSM 17	VK00312	REV 0C	0:0:00:00.00	740-033727
23	1	0	0	jnxMX2020ADC.0	ADC 0	ZV4175	REV 01	0:0:00:00.00	750-043596
23	2	0	0	jnxMX2020ADC.0	ADC 1	ZW1549	REV 02	0:0:00:00.00	750-043596
23	3	0	0	jnxMX2020ADC.0	ADC 2	ZJ0343	REV 01	0:0:00:00.00	750-032317
23	4	0	0	jnxMX2020ADC.0	ADC 3	ZV4091	REV 01	0:0:00:00.00	750-043596
23	5	0	0	jnxMX2020ADC.0	ADC 4	ZW1559	REV 02	0:0:00:00.00	750-043596
23	6	0	0	jnxMX2020ADC.0	ADC 5	ZB2837	REV 01	0:0:00:00.00	750-032317
23	7	0	0	jnxMX2020ADC.0	ADC 6	ZV4097	REV 01	0:0:00:00.00	750-043596
23	8	0	0	jnxMX2020ADC.0	ADC 7	ZE5845	REV 01	0:0:00:00.00	750-032317
23	9	0	0	jnxMX2020ADC.0	ADC 8	ZV4110	REV 01	0:0:00:00.00	750-043596
23	10	0	0	jnxMX2020ADC.0	ADC 9	ZP0752	REV 01	0:0:00:00.00	750-043596
23	11	0	0	jnxMX2020ADC.0	ADC 10	ZV4126	REV 01	0:0:00:00.00	750-043596

Table 56: jnxContentsEntry Objects in the jnxContentsTable of an MX2020 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
23	12	0	0	jnxMX2020ADC.0	ADC 11	ZV4088	REV 01	0:0:00:00.00	750-043596
23	13	0	0	jnxMX2020ADC.0	ADC 12	ZV4096	REV 01	0:0:00:00.00	750-043596
23	14	0	0	jnxMX2020ADC.0	ADC 13	ZV4129	REV 01	0:0:00:00.00	750-043596
23	15	0	0	jnxMX2020ADC.0	ADC 14	ZV4153	REV 01	0:0:00:00.00	750-043596
23	16	0	0	jnxMX2020ADC.0	ADC 15	ZW1566	REV 02	0:0:00:00.00	750-043596
23	17	0	0	jnxMX2020ADC.0	ADC 16	ZV4167	REV 01	0:0:00:00.00	750-043596
23	18	0	0	jnxMX2020ADC.0	ADC 17	ZW1550	REV 02	0:0:00:00.00	750-043596
23	19	0	0	jnxMX2020ADC.0	ADC 18	ZV4152	REV 01	0:0:00:00.00	750-043596
23	20	0	0	jnxMX2020ADC.0	ADC 19	ZV4168	REV 01	0:0:00:00.00	750-043596

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an MX2020 router is as follows:

```

user@host> show chassis hardware
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN11D23B7AFJ  MX2020
Midplane                               ABAC6181      Lower Backplane
Midplane 1                               ABAC4679      Upper Backplane
PMP 1                               ZJ8865        Upper Power Midplane
PMP 0                               ZG6516        Lower Power Midplane
FPM Board                               ZJ8490        Front Panel Display
PSM 5                               VJ00145       DC 52V Power Supply
Module
PSM 6                               REV 0B        740-033727    VG00020        DC 52V Power Supply
Module
PSM 7                               REV 0B        740-033727    VH00063        DC 52V Power Supply
Module
PSM 8                               REV 0B        740-033727    VH00047        DC 52V Power Supply
Module
PSM 14                              REV 0C        740-033727    VJ00140        DC 52V Power Supply
Module
PSM 15                              REV 0B        740-033727    VH00055        DC 52V Power Supply
Module
PSM 16                              REV 0B        740-033727    VH00075        DC 52V Power Supply
Module
PSM 17                              REV 0B        740-033727    VH00091        DC 52V Power Supply
Module
PDM 0                               REV 0A        740-038109    VG00005        DC Power Dist Module
PDM 2                               REV 0A        740-038109    VG00022        DC Power Dist Module
Routing Engine 0 REV 02    740-041821    9009089726    RE-S-1800x4
Routing Engine 1 REV 05    740-031116    9009072774    RE-S-1800x4

```

CB 0	REV 04	750-040257	ZT2845	Control Board
CB 1	REV 00	750-032314	ZE1634	Control Board
SPMB 0	REV 01	711-041855	ZS2288	PMB Board
SPMB 1	REV 06	711-030686	EF3460	SNG PMB
SFB 0	REV 01	711-032385	ZE5866	Switch Fabric Board
SFB 1	REV 01	711-032385	ZE5853	Switch Fabric Board
SFB 2	REV 01	711-032385	ZB7642	Switch Fabric Board
SFB 3	REV 01	711-032385	ZJ3555	Switch Fabric Board
SFB 4	REV 01	711-032385	ZE5850	Switch Fabric Board
SFB 5	REV 01	711-032385	ZE5870	Switch Fabric Board
SFB 6	REV 04	711-032385	ZV4182	Switch Fabric Board
SFB 7	REV 01	711-032385	ZE5858	Switch Fabric Board
FPC 4	REV 09	750-037355	CAAF0937	MPC Type 4-2
CPU	REV 08	711-035209	CAAD8004	HMPC PMB 2G
PIC 0		BUILTIN	BUILTIN	4x10GE SFPP
PIC 1		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0		NON-JNPR	X12J00394	CFP-100G-SR10
PIC 2		BUILTIN	BUILTIN	4x10GE SFPP
Xcvr 0	REV 01	740-031980	153363A00574	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	19T511100399	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AJ50SDT	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AJ102XE	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0	REV 01	740-035329	X12J00042	CFP-100G-SR10
FPC 9	REV 14.3.09	750-033205	YY8443	MPC Type 3
CPU	REV 04	711-035209	YZ0531	HMPC PMB 2G
MIC 0	REV 18	750-028380	YG6885	3D 2x 10GE XFP
PIC 0		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0		NON-JNPR	T09L20487	XFP-10G-SR
PIC 1		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0		NON-JNPR	T10A91821	XFP-10G-SR
MIC 1	REV 07	750-033196	YZ0787	1X100GE CXP
PIC 2		BUILTIN	BUILTIN	1X100GE CXP
Xcvr 0	REV 01	740-032166	XB29FB03F	CXP-100G-SR10
FPC 11	REV 11	750-037358	CAAE2184	MPC Type 4-1
CPU	REV 08	711-035209	CAAE2685	HMPC PMB 2G
PIC 0		BUILTIN	BUILTIN	8X10GE SFPP
Xcvr 0	REV 01	740-021308	19T511100469	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMGOK1S	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	19T511100352	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	193363A00297	SFP+-10G-SR
Xcvr 4	REV 01	740-021308	19T511100498	SFP+-10G-SR
Xcvr 5	REV 01	740-021308	19T511101796	SFP+-10G-SR
Xcvr 6	REV 01	740-021308	19T511101829	SFP+-10G-SR
Xcvr 7	REV 01	740-021308	19T511101603	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	8X10GE SFPP
Xcvr 0	REV 01	740-021308	19T511100373	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	19T511100360	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	T09E07632	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	113363A00171	SFP+-10G-SR
Xcvr 4	REV 01	740-031980	B10L05790	SFP+-10G-SR
Xcvr 5	REV 01	740-021308	T09M83923	SFP+-10G-SR
Xcvr 6	REV 01	740-021308	19T511100261	SFP+-10G-SR
Xcvr 7	REV 01	740-021308	19T511101892	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	8X10GE SFPP
Xcvr 0	REV 01	740-031980	B10M00015	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A02523	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	19T511100493	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11D05630	SFP+-10G-SR
Xcvr 4	REV 01	740-021308	19T511101612	SFP+-10G-SR
Xcvr 5	REV 01	740-031980	183363A02529	SFP+-10G-SR

Xcvr 6	REV 01	740-021308	19T511100950	SFP+-10G-SR
Xcvr 7	REV 01	740-031980	193363A00494	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	8X10GE SFP
Xcvr 0	REV 01	740-021308	AMH014S	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AMG0TVR	SFP+-10G-SR
FPC 16	REV 27	750-033205	ZL6214	MPC Type 3
CPU	REV 07	711-035209	ZK9046	HMPC PMB 2G
MIC 0	REV 10	750-033199	YX4503	1X100GE CFP
PIC 0		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0		NON-JNPR	X12J00384	CFP-100G-SR10
MIC 1	REV 10	750-033199	YX4493	1X100GE CFP
PIC 2		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0		NON-JNPR	X12J00262	CFP-100G-SR10
FPC 17	REV 34	750-031090	ZT7111	MPC Type 2 3D EQ
CPU	REV 06	711-030884	ZT7719	MPC PMB 2G
MIC 0	REV 26	750-028392	ZT8724	3D 20x 1GE(LAN) SFP
PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 01	740-011782	P8Q2318	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 01	740-011613	PE70XRB	SFP-SX
Xcvr 1	REV 01	740-021309	91D104A00011	UNSUPPORTED
MIC 1	REV 27	750-028387	ZW4705	3D 4x 10GE XFP
PIC 2		BUILTIN	BUILTIN	2x 10GE XFP
Xcvr 0		NON-JNPR	T09L25570	XFP-10G-SR
Xcvr 1		NON-JNPR	T09K81549	XFP-10G-SR
PIC 3		BUILTIN	BUILTIN	2x 10GE XFP
Xcvr 0	REV 02	740-014289	T09G64335	XFP-10G-SR
QXM 0	REV 06	711-028408	ZT4307	MPC QXM
QXM 1	REV 06	711-028408	ZT3975	MPC QXM
FPC 18	REV 30	750-028467	ZM4986	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ZP6541	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	19T511101842	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B10M00261	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	19T511100363	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02732	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 3	REV 01	740-021308	19T511101609	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A03065	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	T09E07837	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A03046	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	19T511101801	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	19T511100643	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	19T511100646	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	19T511101008	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	19T511101816	SFP+-10G-SR
ADC 4	REV 05	750-043596	CAAC2078	Adapter Card
ADC 9	REV 01	750-032317	ZB2836	ADC Etch1
ADC 11	REV 01	750-043596	ZV4079	Adapter Card
ADC 16	REV 05	750-043596	CAAC2057	Adapter Card
ADC 17	REV 05	750-043596	CAAC2064	Adapter Card
ADC 18	REV 05	750-043596	CAAC2077	Adapter Card
Fan Tray 0	REV 1A	760-042349	VJ00001	Fan Tray
Fan Tray 1	REV 1A	760-042349	VJ00002	Fan Tray
Fan Tray 2	REV 1A	760-042349	VJ00015	Fan Tray
Fan Tray 3	REV 1A	760-042349	VJ00006	Fan Tray

Table 57 on page 162 provides an example of `jnxContentEntry` objects in the `jnxContentTable` of a T640 router.

Table 57: 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	—
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	—

Table 57: 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	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
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

Table 57: 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
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
10	1	0	0	jnxT640FPB.0	FPM	HE3245	REV 02	0:0:00:00.00	710-002901
11	1	0	0	jnxT640SCG.0	SCG 0	HF6023	REV 04	0:0:00:00.00	710-003423
11	2	0	0	jnxT640SCG.0	SCG 1	HF6061	REV 04	0:0:00:00.00	710-003423
12	2	0	0	jnxT640CB.0	CB 0	HE3614	REV 06	0:0:00:00.00	710-002728
12	2	0	0	jnxT640CB.0	CB 1	HE3627	REV 06	0:0:00:00.00	710-002728
13	1	0	0	jnxT640CIP.0	CIP	HA4729	REV 05	0:0:00:00.00	710-002895

Table 57: 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
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 58 on page 166 provides an example of `jnxContentEntry` objects in the `jnxContentTable` of a T320 router.

Table 58: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	<code>jnxMidplaneT320.0</code>	Midplane	AY4527	Rev 01	(0) 0:00:00.00	710-004339
2	1	0	0	<code>jnxT320Power.0</code>	PEM 0	ML14099	Rev 01	(0) 0:00:00.00	—
4	1	1	0	<code>jnxT320Fan.0</code>	Top left front fan	—	—	(0) 0:00:00.00	—
4	1	2	0	<code>jnxT320Fan.0</code>	Top left middle fan	—	—	(0) 0:00:00.00	—
4	1	3	0	<code>jnxT320Fan.0</code>	Top left rear fan	—	—	(0) 0:00:00.00	—
4	1	4	0	<code>jnxT320Fan.0</code>	Top right front fan	—	—	(0) 0:00:00.00	—
4	1	5	0	<code>jnxT320Fan.0</code>	Top right middle fan	—	—	(0) 0:00:00.00	—
4	1	6	0	<code>jnxT320Fan.0</code>	Top right rear fan	—	—	(0) 0:00:00.00	—
4	2	1	0	<code>jnxT320Fan.0</code>	Bottom left front fan	—	—	(0) 0:00:00.00	—
4	2	2	0	<code>jnxT320Fan.0</code>	Bottom left middle fan	—	—	(0) 0:00:00.00	—

Table 58: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router (continued)

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
4	2	3	0	jnxT320Fan.0	Bottom left rear fan	—	—	(0) 0:00:00.00	—
4	2	4	0	jnxT320Fan.0	Bottom right front fan	—	—	(0) 0:00:00.00	—
4	2	5	0	jnxT320Fan.0	Bottom right middle fan	—	—	(0) 0:00:00.00	—
4	2	6	0	jnxT320Fan.0	Bottom right rear fan	—	—	(0) 0:00:00.00	—
4	3	1	0	jnxT320Fan.0	Rear tray top fan	—	—	(0) 0:00:00.00	—
4	3	2	0	jnxT320Fan.0	Rear tray second fan	—	—	(0) 0:00:00.00	—
4	3	3	0	jnxT320Fan.0	Rear tray middle fan	—	—	(0) 0:00:00.00	—
4	3	4	0	jnxT320Fan.0	Rear tray fourth fan	—	—	(0) 0:00:00.00	—
4	3	5	0	jnxT320Fan.0	Rear tray bottom fan	—	—	(0) 0:00:00.00	—
7	4	0	0	jnxT320FPC.0	FPC @ 3/*/*	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	1	0	jnxT320FPC.0	FPC @ 3/0/* top 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 SMSR2 @ 0/1/*	HJ9298	REV 06	(6434) 0:01:04.34	750-004535

Table 58: 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
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
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 59 on page 169 provides an example of `jnxContentstEntry` objects in the `jnxContentTable` of a T4000 router.

Table 59: 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	<code>jnxMidplaneT1600.0</code>	Midplane	SNR0652	REV 01	0:0:00:00.00	710-027486
2	1	0	0	<code>jnxPower.0</code>	PEM 0	MD21815	REV 01	0:0:00:00.00	40-026384
4	1	0	0	<code>jnxT4000Fan.0</code>	Front Top Fan Tray	—	—	0:0:00:00.00	—
4	1	1	0	<code>jnxT4000Fan.0</code>	Top Left Front fan	—	—	0:0:00:00.00	—
4	1	2	0	<code>jnxT4000Fan.0</code>	Top Left Middle fan	—	—	0:0:00:00.00	—
4	1	3	0	<code>jnxT4000Fan.0</code>	Top Left Rear fan	—	—	0:0:00:00.00	—
4	1	4	0	<code>jnxT4000Fan.0</code>	Top Right Front fan	—	—	0:0:00:00.00	—
4	1	5	0	<code>jnxT4000Fan.0</code>	Top right Middle fan	—	—	0:0:00:00.00	—

Table 59: 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	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	–
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	–

Table 59: 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	8	0	jnxT4000Fan.0	Rear Tray Bottom Fan	—	—	0:0:00:00.00	—
7	2	0	0	jnxT4000FPC.0	FPC Type 4-ES@ 1/*/*	SNE369	REV 03	0:18:57:02.71	710-033871
7	2	1	0	jnxT4000FPC.0	FPC Type 4-ES@ 1/*/* top temperature sensor	SNE369	REV 03	0:18:57:02.71	710-033871
7	2	2	0	jnxT4000FPC.0	FPC Type 4-ES@ 1/*/* bottom temperature sensor	SNE369	REV 03	0:18:57:02.71	710-033871
7	3	0	0	jnxT4000FPC.0	FPC Type 5-3D@ 2/*/*	SNE366	REV 07	0:18:56:52.85	750-010153
7	3	1	0	jnxT4000FPC.0	FPC Type 5-3D@ 2/*/*	SNE366	REV 07	0:18:56:52.85	750-010153
7	3	2	0	jnxT4000FPC.0	FPC Type 5-3D@ 2/*/* exhaust	SNE366	REV 07	0:18:56:52.85	750-010153
7	5	0	0	jnxT4000FPC.0	FPC Type 5-3D @ 4/*/*	SNE364	REV 07	0:18:56:52.85	750-032819
7	5	1	0	jnxT4000FPC.0	FPC Type 5-3D @ 4/*/* intake	SNE364	REV 07	0:18:56:52.85	750-032819
7	5	2	0	jnxT4000FPC.0	FPC Type 5-3D @ 4/*/* exhaust	SNE364	REV 07	0:18:56:52.85	750-032819
8	3	1	0	jnxT4000PIC3.0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/0/*	SNE374	REV 02	0:18:57:04.47	50-034624
8	3	2	0	jnxT4000PIC3.0	PIC: 12x10GE (LAN/WAN) SFPP @ 2/1/*	SNE375	REV 02	0:18:57:04.55	750-034624
8	5	1	0	jnxT4000PIC3.0	PIC: 12x10GE (LAN/WAN) SFPP @ 4/1/*	SNE378	REV 02	0:18:56:55.16	750-034624

Table 59: 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
9	1	2	0	jnxT4000HM.0	Routing Engine 0	PAF0025	REV 06	0:19:19:30.95	740-02694
9	2	0	0	jnxT4000HM.0	Routing Engine 1	PAF0025	REV 06	2:19:45:51.00	740-02694
10	1	1	0	jnxT4000FPB.0	FPM GBUS	SBAE0927	REV 13	0:0:00:00.00	710-002901
10	1	2	0	jnxT4000FPB.0	FPM Display	SBAE0927	REV 13	0:0:00:00.00	710-021387
11	1	0	0	jnxT4000SCG.0	SCG 0	BBAD9210	REV 18	0:0:00:00.00	710-003423
11	2	0	0	jnxT4000SCG.0	SCG 1	BBAD9210	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	SNEF7358	REV 09	0:0:00:00.00	710-022597
13	1	0	0	jnxT4000CIP.0	CIP	BBAD9210	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	SNEF6201	REV 02	0:0:00:00.00	710-023321
15	2	0	0	jnxT4000SIB.0	SIB 1	SNEF6202	REV 07	0:0:00:00.00	711-036340
15	3	0	0	jnxT4000SIB.0	SIB 2	SNEF6308	REV 02	0:0:00:00.00	711-036340
15	4	0	0	jnxT4000SIB.0	SIB 3	SNEF6317	REV 02	0:0:00:00.00	711-036340
15	5	0	0	jnxT4000SIB.0	SIB 4	SNEF6301	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
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 **jnxLEDTTable** 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.
- **jnxLEDOrganator**—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 60 on page 176 through Table 64 on page 183 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—**jnxLEDOrganizer**
- Description—**jnxLEDDescr**
- State—**jnxLEDState**

Table 60 on page 176 provides an example of **jnxLEDEntry** objects in the **jnxLEDTable** of an M20 router.

Table 60: 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 61 on page 177 provides an example of `jnxLEDEntry` objects in the `jnxLEDTable` of an MX2010 router.

Table 61: `jnxLEDEntry` Objects in the `jnxLEDTable` of an MX2010 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
<code>jnxContentsTable</code>	1	1	0	0	0.0	Chassis alarm LED	Other
<code>jnxContentsTable</code>	7	1	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 0 LED	Green
<code>jnxContentsTable</code>	7	2	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 1 LED	Green
<code>jnxContentsTable</code>	7	3	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 2 LED	Other
<code>jnxContentsTable</code>	7	4	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 3 LED	Other
<code>jnxContentsTable</code>	7	5	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 4 LED	Other
<code>jnxContentsTable</code>	7	6	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 5 LED	Other
<code>jnxContentsTable</code>	7	7	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 6 LED	Other
<code>jnxContentsTable</code>	7	8	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 7 LED	Other
<code>jnxContentsTable</code>	7	9	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 8 LED	Green
<code>jnxContentsTable</code>	7	10	0	0	<code>jnxMX2010FPC.0</code>	FPC slot 9 LED	Green
<code>jnxContentsTable</code>	9	1	0	0	<code>jnxMX2010HM.0</code>	Routing Engine 0 LED	Green
<code>jnxContentsTable</code>	9	2	0	0	<code>jnxMX2010HM.0</code>	Routing Engine 1 LED	Other
<code>jnxContentsTable</code>	12	1	0	0	<code>jnxMX2010CB.0</code>	CB slot 0 LED	Blue
<code>jnxContentsTable</code>	12	2	0	0	<code>jnxMX2010CB.0</code>	CB slot 1 LED	Other
<code>jnxContentsTable</code>	14	1	0	0	<code>jnxMX2010SPMB.0</code>		Other
<code>jnxContentsTable</code>	14	2	0	0	<code>jnxMX2010SPMB.0</code>		Other
<code>jnxContentsTable</code>	15	1	0	0	<code>jnxMX2010SFB.0</code>	SFB slot 0 LED	Other
<code>jnxContentsTable</code>	15	2	0	0	<code>jnxMX2010SFB.0</code>	SFB slot 1 LED	Other
<code>jnxContentsTable</code>	15	3	0	0	<code>jnxMX2010SFB.0</code>	SFB slot 2 LED	Other
<code>jnxContentsTable</code>	15	4	0	0	<code>jnxMX2010SFB.0</code>	SFB slot 3 LED	Other

Table 61: jnxLEDEntry Objects in the jnxLEDTable of an MX2010 Router (*continued*)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	15	5	0	0	jnxMX2010SFB.0	SFB slot 4 LED	Other
jnxContentsTable	15	6	0	0	jnxMX2010SFB.0	SFB slot 5 LED	Other
jnxContentsTable	15	7	0	0	jnxMX2010SFB.0	SFB slot 6 LED	Other
jnxContentsTable	15	8	0	0	jnxMX2010SFB.0	SFB slot 7 LED	Other
jnxContentsTable	21	1	0	0	jnxMX2010PDM.0	PDM 0 LED	Green
jnxContentsTable	21	2	0	0	jnxMX2010PDM.0	PDM 1 LED	Green
jnxContentsTable	22	1	0	0	jnxMX2010PSM.0	PSM 0 LED	Other
jnxContentsTable	22	2	0	0	jnxMX2010PSM.0	PSM 1 LED	Other
jnxContentsTable	22	3	0	0	jnxMX2010PSM.0	PSM 2 LED	Other
jnxContentsTable	22	4	0	0	jnxMX2010PSM.0	PSM 3 LED	Other
jnxContentsTable	22	5	0	0	jnxMX2010PSM.0	PSM 4 LED	Green
jnxContentsTable	22	6	0	0	jnxMX2010PSM.0	PSM 5 LED	Green
jnxContentsTable	22	7	0	0	jnxMX2010PSM.0	PSM 6 LED	Green
jnxContentsTable	22	8	0	0	jnxMX2010PSM.0	PSM 7 LED	Green
jnxContentsTable	22	9	0	0	jnxMX2010PSM.0	PSM 8 LED	Green
jnxContentsTable	23	1	0	0	jnxMX2010ADC.0	ADC slot 0 LED	Other
jnxContentsTable	23	2	0	0	jnxMX2010ADC.0	ADC slot 1 LED	Other
jnxContentsTable	23	3	0	0	jnxMX2010ADC.0	ADC slot 2 LED	Other
jnxContentsTable	23	4	0	0	jnxMX2010ADC.0	ADC slot 3 LED	Other
jnxContentsTable	23	5	0	0	jnxMX2010ADC.0	ADC slot 4 LED	Other
jnxContentsTable	23	6	0	0	jnxMX2010ADC.0	ADC slot 5 LED	Other
jnxContentsTable	23	7	0	0	jnxMX2010ADC.0	ADC slot 6 LED	Other
jnxContentsTable	23	8	0	0	jnxMX2010ADC.0	ADC slot 7 LED	Other

Table 61: jnxLEDEntry Objects in the jnxLEDTable of an MX2010 Router (*continued*)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	23	9	0	0	jnxMX2010ADC.0	ADC slot 8 LED	Other
jnxContentsTable	23	10	0	0	jnxMX2010ADC.0	ADC slot 9 LED	Other

Table 62 on page 179 provides an example of jnxLEDEntry objects in the jnxLEDTable of an MX2020 router.

Table 62: jnxLEDEntry Objects in the jnxLEDTable of an MX2020 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	0.0	Chassis alarm LED	Other
jnxContentsTable	4	1	0	0	jnxMX2020Fan.0	FAN 0 LED	Other
jnxContentsTable	4	2	0	0	jnxMX2020Fan.0	FAN 1 LED	Other
jnxContentsTable	4	3	0	0	jnxMX2020Fan.0	FAN 2 LED	Other
jnxContentsTable	4	4	0	0	jnxMX2020Fan.0	FAN 3 LED	Other
jnxContentsTable	7	1	0	0	jnxMX2020FPC.0	FPC slot 0 LED	Blue
jnxContentsTable	7	2	0	0	jnxMX2020FPC.0	FPC slot 1 LED	Other
jnxContentsTable	7	3	0	0	jnxMX2020FPC.0	FPC slot 2 LED	Other
jnxContentsTable	7	4	0	0	jnxMX2020FPC.0	FPC slot 3 LED	Other
jnxContentsTable	7	5	0	0	jnxMX2020FPC.0	FPC slot 4 LED	Other
jnxContentsTable	7	6	0	0	jnxMX2020FPC.0	FPC slot 5 LED	Other
jnxContentsTable	7	7	0	0	jnxMX2020FPC.0	FPC slot 6 LED	Other
jnxContentsTable	7	8	0	0	jnxMX2020FPC.0	FPC slot 7 LED	Other
jnxContentsTable	7	9	0	0	jnxMX2020FPC.0	FPC slot 8 LED	Other
jnxContentsTable	7	10	0	0	jnxMX2020FPC.0	FPC slot 9 LED	Other
jnxContentsTable	7	11	0	0	jnxMX2020FPC.0	FPC slot 10 LED	Other
jnxContentsTable	7	12	0	0	jnxMX2020FPC.0	FPC slot 11 LED	Other

Table 62: jnxLEDEntry Objects in the jnxLEDTable of an MX2020 Router (*continued*)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	7	13	0	0	jnxMX2020FPC.0	FPC slot 12 LED	Other
jnxContentsTable	7	14	0	0	jnxMX2020FPC.0	FPC slot 13 LED	Other
jnxContentsTable	7	15	0	0	jnxMX2020FPC.0	FPC slot 14 LED	Other
jnxContentsTable	7	16	0	0	jnxMX2020FPC.0	FPC slot 15 LED	Other
jnxContentsTable	7	17	0	0	jnxMX2020FPC.0	FPC slot 16 LED	Other
jnxContentsTable	7	18	0	0	jnxMX2020FPC.0	FPC slot 17 LED	Other
jnxContentsTable	7	19	0	0	jnxMX2020FPC.0	FPC slot 18 LED	Other
jnxContentsTable	7	20	0	0	jnxMX2020FPC.0	FPC slot 19 LED	Other
jnxContentsTable	9	1	0	0	jnxMX2020HM.0	Routing Engine 0 LED	Green
jnxContentsTable	9	2	0	0	jnxMX2020HM.0	Routing Engine 1 LED	Green
jnxContentsTable	12	1	0	0	jnxMX2020CB.0	CB slot 0 LED	Blue
jnxContentsTable	12	2	0	0	jnxMX2020CB.0	CB slot 1 LED	Green
jnxContentsTable	14	1	0	0	jnxMX2020SPMB.0		Other
jnxContentsTable	14	2	0	0	jnxMX2020SPMB.0		Other
jnxContentsTable	15	1	0	0	jnxMX2020SFB.0	SFB slot 0 LED	Other
jnxContentsTable	15	2	0	0	jnxMX2020SFB.0	SFB slot 1 LED	Other
jnxContentsTable	15	3	0	0	jnxMX2020SFB.0	SFB slot 2 LED	Other
jnxContentsTable	15	4	0	0	jnxMX2020SFB.0	SFB slot 3 LED	Other
jnxContentsTable	15	5	0	0	jnxMX2020SFB.0	SFB slot 4 LED	Other
jnxContentsTable	15	6	0	0	jnxMX2020SFB.0	SFB slot 5 LED	Other
jnxContentsTable	15	7	0	0	jnxMX2020SFB.0	SFB slot 6 LED	Other
jnxContentsTable	15	8	0	0	jnxMX2020SFB.0	SFB slot 7 LED	Other
jnxContentsTable	21	1	0	0	jnxMX2020PDM.0	PDM 0 LED	Green

Table 62: jnxLEDEntry Objects in the jnxLEDTable of an MX2020 Router (*continued*)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	21	2	0	0	jnxMX2020PDM.0	PDM 1 LED	Other
jnxContentsTable	21	3	0	0	jnxMX2020PDM.0	PDM 2 LED	Green
jnxContentsTable	21	4	0	0	jnxMX2020PDM.0	PDM 3 LED	Other
jnxContentsTable	22	1	0	0	jnxMX2020PSM.0	PSM 0 LED	Other
jnxContentsTable	22	2	0	0	jnxMX2020PSM.0	PSM 1 LED	Other
jnxContentsTable	22	3	0	0	jnxMX2020PSM.0	PSM 2 LED	Other
jnxContentsTable	22	4	0	0	jnxMX2020PSM.0	PSM 3 LED	Other
jnxContentsTable	22	5	0	0	jnxMX2020PSM.0	PSM 4 LED	Other
jnxContentsTable	22	6	0	0	jnxMX2020PSM.0	PSM 5 LED	Other
jnxContentsTable	22	7	0	0	jnxMX2020PSM.0	PSM 6 LED	Green
jnxContentsTable	22	8	0	0	jnxMX2020PSM.0	PSM 7 LED	Red
jnxContentsTable	22	9	0	0	jnxMX2020PSM.0	PSM 8 LED	Green
jnxContentsTable	22	10	0	0	jnxMX2020PSM.0	PSM 9 LED	Other
jnxContentsTable	22	11	0	0	jnxMX2020PSM.0	PSM 10 LED	Green
jnxContentsTable	22	12	0	0	jnxMX2020PSM.0	PSM 11 LED	Other
jnxContentsTable	22	13	0	0	jnxMX2020PSM.0	PSM 12 LED	Other
jnxContentsTable	22	14	0	0	jnxMX2020PSM.0	PSM 13 LED	Other
jnxContentsTable	22	15	0	0	jnxMX2020PSM.0	PSM 14 LED	Red
jnxContentsTable	22	16	0	0	jnxMX2020PSM.0	PSM 15 LED	Other
jnxContentsTable	22	17	0	0	jnxMX2020PSM.0	PSM 16 LED	Other
jnxContentsTable	22	18	0	0	jnxMX2020PSM.0	PSM 17 LED	Red
jnxContentsTable	23	1	0	0	jnxMX2020ADC.0	ADC slot 0 LED	Other
jnxContentsTable	23	2	0	0	jnxMX2020ADC.0	ADC slot 1 LED	Other

Table 62: jnxLEDEntry Objects in the jnxLEDTable of an MX2020 Router (continued)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	23	3	0	0	jnxMX2020ADC.0	ADC slot 2 LED	Other
jnxContentsTable	23	4	0	0	jnxMX2020ADC.0	ADC slot 3 LED	Other
jnxContentsTable	23	5	0	0	jnxMX2020ADC.0	ADC slot 4 LED	Other
jnxContentsTable	23	6	0	0	jnxMX2020ADC.0	ADC slot 5 LED	Other
jnxContentsTable	23	7	0	0	jnxMX2020ADC.0	ADC slot 6 LED	Other
jnxContentsTable	23	8	0	0	jnxMX2020ADC.0	ADC slot 7 LED	Other
jnxContentsTable	23	9	0	0	jnxMX2020ADC.0	ADC slot 8 LED	Other
jnxContentsTable	23	10	0	0	jnxMX2020ADC.0	ADC slot 9 LED	Other
jnxContentsTable	23	11	0	0	jnxMX2020ADC.0	ADC slot 10 LED	Other
jnxContentsTable	23	12	0	0	jnxMX2020ADC.0	ADC slot 11 LED	Other
jnxContentsTable	23	13	0	0	jnxMX2020ADC.0	ADC slot 12 LED	Other
jnxContentsTable	23	14	0	0	jnxMX2020ADC.0	ADC slot 13 LED	Other
jnxContentsTable	23	15	0	0	jnxMX2020ADC.0	ADC slot 14 LED	Other
jnxContentsTable	23	16	0	0	jnxMX2020ADC.0	ADC slot 15 LED	Other
jnxContentsTable	23	17	0	0	jnxMX2020ADC.0	ADC slot 16 LED	Other
jnxContentsTable	23	18	0	0	jnxMX2020ADC.0	ADC slot 17 LED	Other
jnxContentsTable	23	19	0	0	jnxMX2020ADC.0	ADC slot 18 LED	Other
jnxContentsTable	23	20	0	0	jnxMX2020ADC.0	ADC slot 19 LED	Other

Table 63 on page 182 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T640 router.

Table 63: 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

Table 63: 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	1	0	0	jnxT640FPC.0	FPC slot 0 LED	Other
jnxContentsTable	7	2	0	0	jnxT640FPC.0	FPC slot 1 LED	Green
jnxContentsTable	7	3	0	0	jnxT640FPC.0	FPC slot 2 LED	Other
jnxContentsTable	7	4	0	0	jnxT640FPC.0	FPC slot 3 LED	Other
jnxContentsTable	7	5	0	0	jnxT640FPC.0	FPC slot 4 LED	Other
jnxContentsTable	7	6	0	0	jnxT640FPC.0	FPC slot 5 LED	Green
jnxContentsTable	7	7	0	0	jnxT640FPC.0	FPC slot 6 LED	Other
jnxContentsTable	7	8	0	0	jnxT640FPC.0	FPC slot 7 LED	Green
jnxContentsTable	9	1	0	0	jnxT640HM.0	Host 0 LED	Blue
jnxContentsTable	9	2	0	0	jnxT640HM.0	Host 1 LED	Green

Table 64 on page 183 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T320 router.

Table 64: 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

Table 64: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router (*continued*)

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable(3)	9	1	0	0	jnxT320HM.0	Host 0 LED	Blue
jnxContentsTable(3)	9	2	0	0	jnxT320HM.0	Host 1 LED	Green

jnxFilledLastChange

The object identifier for the **jnxFilledLastChange** object is **{jnxBoxAnatomy 11}**. This object indicates when the box filled status last changed. This variable is 0 if the **sysUpTime** value is unknown or it already existed when the agent was active.

jnxFilledTable

The object identifier for the **jnxFilledTable** object is **{jnxBoxAnatomy 12}**. This object indicates whether a specific container in the router is used (filled) or empty. This table is used for inventory and capacity planning.

Entries in the **jnxFilledTable** are represented by the **jnxFilledEntry** object, whose object identifier is **{jnxFilledTable 1}**.

The **jnxFilledTable** describes the status of specific containers whose component objects are described as follows:

- **jnxFilledContainerIndex**—The associated **jnxContainersIndex** in the **jnxContainersTable**, whose object identifier is **{jnxFilledEntry 1}**.
- **jnxFilledL1Index**—The level-one index of the container housing the entry, whose object identifier is **{jnxFilledEntry 2}**.
- **jnxFilledL2Index**—The level-two index of the container housing the entry, whose object identifier is **{jnxFilledEntry 3}**.
- **jnxFilledL3Index**—The level-three index of the container housing the entry, whose object identifier is **{jnxFilledEntry 4}**.
- **jnxFilledDescr**—The entry's name or detailed description of the entry, whose object identifier is **{jnxFilledEntry 5}**.
- **jnxFilledState**—The entry's state (filled or empty), whose object identifier is **{jnxFilledEntry 6}**.

Table 65 on page 185 through Table 69 on page 207 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 65 on page 185 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of an M20 router.

Table 65: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame compartment	Filled
1	1	1	0	Temperature sensor space 0	Filled
1	1	2	0	Temperature sensor space 1	Filled
2	1	0	0	Power supply compartment A	Filled
2	2	0	0	Power supply compartment B	Empty
3	1	0	0	Rear top impeller compartment	Filled
3	2	0	0	Front bottom impeller compartment	Filled
4	1	0	0	Rear left fan compartment	Filled
4	2	0	0	Right center fan compartment	Filled
4	3	0	0	Rear right fan compartment	Filled
5	1	0	0	Host controller compartment	Filled
6	1	0	0	SCB slot	Filled
7	1	0	0	FPC slot 0	Empty
7	2	0	0	FPC slot 1	Empty
7	3	0	0	FPC slot 2	Filled
7	4	0	0	FPC slot 3	Filled
7	5	0	0	FPC slot 4	Empty
7	6	0	0	FPC slot 5	Filled
7	7	0	0	FPC slot 6	Empty

Table 65: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router (continued)

Container Index	L1	L2	L3	Description	State
7	8	0	0	FPC slot 7	Empty
8	1	1	0	PIC space @ 0/0/*	Empty
8	1	2	0	PIC space @ 0/1/*	Empty
8	1	3	0	PIC space @ 0/2/*	Empty
8	1	4	0	PIC space @ 0/3/*	Empty
8	2	1	0	PIC space @ 1/0/*	Empty
8	2	2	0	PIC space @ 1/1/*	Empty
8	2	3	0	PIC space @ 1/2/*	Empty
8	2	4	0	PIC space @ 1/3/*	Empty
8	3	1	0	PIC space @ 2/0/*	Filled
8	3	2	0	PIC space @ 2/1/*	Filled
8	3	3	0	PIC space @ 2/2/*	Filled
8	3	4	0	PIC space @ 2/3/*	Filled
8	4	1	0	PIC space @ 3/0/*	Filled
8	4	2	0	PIC space @ 3/1/*	Filled
8	4	3	0	PIC space @ 3/2/*	Filled
8	4	4	0	PIC space @ 3/3/*	Filled
8	5	1	0	PIC space @ 4/0/*	Empty
8	5	2	0	PIC space @ 4/1/*	Empty
8	5	3	0	PIC space @ 4/2/*	Empty
8	5	4	0	PIC space @ 4/3/*	Empty
8	6	1	0	PIC space @ 5/0/*	Filled
8	6	2	0	PIC space @ 5/1/*	Filled

Table 65: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router (continued)

Container Index	L1	L2	L3	Description	State
8	6	3	0	PIC space @ 5/2/*	Filled
8	6	4	0	PIC space @ 5/3/*	Filled
8	7	1	0	PIC space @ 6/0/*	Empty
8	7	2	0	PIC space @ 6/1/*	Empty
8	7	3	0	PIC space @ 6/2/*	Empty
8	7	4	0	PIC space @ 6/3/*	Empty
8	8	1	0	PIC space @ 7/0/*	Empty
8	8	2	0	PIC space @ 7/1/*	Empty
8	8	3	0	PIC space @ 7/2/*	Empty
8	8	4	0	PIC space @ 7/3/*	Empty
9	1	0	0	Routing Engine compartment	Filled

Table 66 on page 187 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of an MX2010 router.

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
4	1	1	0	Fan Tray 0 Fan 1 slot	Filled
4	1	2	0	Fan Tray 0 Fan 2 slot	Filled
4	1	3	0	Fan Tray 0 Fan 3 slot	Filled
4	1	4	0	Fan Tray 0 Fan 4 slot	Filled
4	1	5	0	Fan Tray 0 Fan 5 slot	Filled
4	1	6	0	Fan Tray 0 Fan 6 slot	Filled
4	2	1	0	Fan Tray 1 Fan 1 slot	Filled

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
4	2	2	0	Fan Tray 1 Fan 2 slot	Filled
4	2	3	0	Fan Tray 1 Fan 3 slot	Filled
4	2	4	0	Fan Tray 1 Fan 4 slot	Filled
4	2	5	0	Fan Tray 1 Fan 5 slot	Filled
4	2	6	0	Fan Tray 1 Fan 6 slot	Filled
4	3	1	0	Fan Tray 2 Fan 1 slot	Filled
4	3	2	0	Fan Tray 2 Fan 2 slot	Filled
4	3	3	0	Fan Tray 2 Fan 3 slot	Filled
4	3	4	0	Fan Tray 2 Fan 4 slot	Filled
4	3	5	0	Fan Tray 2 Fan 5 slot	Filled
4	3	6	0	Fan Tray 2 Fan 6 slot	Filled
4	4	1	0	Fan Tray 3 Fan 1 slot	Filled
4	4	2	0	Fan Tray 3 Fan 2 slot	Filled
4	4	3	0	Fan Tray 3 Fan 3 slot	Filled
4	4	4	0	Fan Tray 3 Fan 4 slot	Filled
4	4	5	0	Fan Tray 3 Fan 5 slot	Filled
4	4	6	0	Fan Tray 3 Fan 6 slot	Filled
7	1	0	0	FPC slot 0	Filled
7	2	0	0	FPC slot 1	Filled
7	3	0	0	FPC slot 2	Empty
7	4	0	0	FPC slot 3	Empty
7	5	0	0	FPC slot 4	Empty
7	6	0	0	FPC slot 5	Empty

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
7	7	0	0	FPC slot 6	Empty
7	8	0	0	FPC slot 7	Empty
7	9	0	0	FPC slot 8	Filled
7	10	0	0	FPC slot 9	Filled
8	1	1	0	PIC slot @ 0/0/*	Filled
8	1	2	0	PIC slot @ 0/1/*	Filled
8	1	3	0	PIC slot @ 0/2/*	Filled
8	1	4	0	PIC slot @ 0/3/*	Filled
8	2	1	0	PIC slot @ 1/0/*	Filled
8	2	2	0	PIC slot @ 1/1*	Empty
8	2	3	0	PIC slot @ 1/2/*	Filled
8	2	4	0	PIC slot @ 1/3/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	3	3	0	PIC slot @ 2/2/*	Empty
8	3	4	0	PIC slot @ 2/3/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Empty
8	4	2	0	PIC slot @ 3/1/*	Empty
8	4	3	0	PIC slot @ 3/2/*	Empty
8	4	4	0	PIC slot @ 3/3/*	Empty
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	3	0	PIC slot @ 4/2/*	Empty

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
8	5	4	0	PIC slot @ 4/3/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Empty
8	6	2	0	PIC slot @ 5/1/*	Empty
8	6	3	0	PIC slot @ 5/2/*	Empty
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/*	Empty
8	8	2	0	PIC slot @ 7/1/*	Empty
8	8	3	0	PIC slot @ 7/2/*	Empty
8	8	4	0	PIC slot @ 7/3/*	Empty
8	9	1	0	PIC slot @ 8/0/*	Filled
8	9	2	0	PIC slot @ 8/1/*	Filled
8	9	3	0	PIC slot @ 8/2/*	Filled
8	9	4	0	PIC slot @ 8/3/*	Filled
8	10	1	0	PIC slot @ 9/0/*	Filled
8	10	2	0	PIC slot @ 9/1/*	Filled
8	10	3	0	PIC slot @ 9/2/*	Filled
8	10	4	0	PIC slot @ 9/3/*	Filled
9	1	0	0	Routing Engine 0	Filled
9	2	0	0	Routing Engine 1	Filled

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
10	1	1	0	FPM Board slot	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SFB slot 0	Filled
15	2	0	0	SFB slot 1	Filled
15	3	0	0	SFB slot 2	Filled
15	4	0	0	SFB slot 3	Filled
15	5	0	0	SFB slot 4	Filled
15	6	0	0	SFB slot 5	Filled
15	7	0	0	SFB slot 6	Filled
15	8	0	0	SFB slot 7	Filled
20	1	1	0	MIC slot @ 0/0/*	Empty
20	1	2	0	MIC slot @ 0/1/*	Empty
20	2	1	0	MIC slot @ 1/0/*	Filled
20	2	2	0	MIC slot @ 1/1/*	Filled
20	3	1	0	MIC slot @ 2/0/*	Empty
20	3	2	0	MIC slot @ 2/1/*	Empty
20	4	1	0	MIC slot @ 3/0/*	Empty
20	4	2	0	MIC slot @ 3/1/*	Empty
20	5	1	0	MIC slot @ 4/0/*	Empty
20	5	2	0	MIC slot @ 4/1/*	Empty

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
20	6	1	0	MIC slot @ 5/0/*	Empty
20	6	2	0	MIC slot @ 5/1/*	Empty
20	7	1	0	MIC slot @ 6/0/*	Empty
20	7	2	0	MIC slot @ 6/1/*	Empty
20	8	1	0	MIC slot @ 7/0/*	Empty
20	8	2	0	MIC slot @ 7/1/*	Empty
20	9	1	0	MIC slot @ 8/0/*	Filled
20	9	2	0	MIC slot @ 8/1/*	Filled
20	10	1	0	MIC slot @ 9/0/*	Empty
20	10	2	0	MIC slot @ 9/1/*	Empty
21	1	0	0	PDM slot 0	Filled
21	2	0	0	PDM slot 1	Filled
22	1	0	0	PSM slot 0	Empty
22	2	0	0	PSM slot 1	Empty
22	3	0	0	PSM slot 2	Empty
22	4	0	0	PSM slot 3	Empty
22	5	0	0	PSM slot 4	Filled
22	6	0	0	PSM slot 5	Filled
22	7	0	0	PSM slot 6	Filled
22	8	0	0	PSM slot 7	Filled
22	9	0	0	PSM slot 8	Filled
23	1	0	0	ADC slot 0	Filled
23	2	0	0	ADC slot 1	Filled

Table 66: jnxFilledEntry Objects in the jnxFilledTable of an MX2010 Router (continued)

Container Index	L1	L2	L3	Description	State
23	3	0	0	ADC slot 2	Empty
23	4	0	0	ADC slot 3	Empty
23	5	0	0	ADC slot 4	Empty
23	6	0	0	ADC slot 5	Empty
23	7	0	0	ADC slot 6	Empty
23	8	0	0	ADC slot 7	Empty
23	9	0	0	ADC slot 8	Filled
23	10	0	0	ADC slot 9	Filled

[Table 67 on page 193](#) provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of an MX2020 router.

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
4	1	1	0	Fan Tray 0 Fan 1 slot	Filled
4	1	2	0	Fan Tray 0 Fan 2 slot	Filled
4	1	3	0	Fan Tray 0 Fan 3 slot	Filled
4	1	4	0	Fan Tray 0 Fan 4 slot	Filled
4	1	5	0	Fan Tray 0 Fan 5 slot	Filled
4	1	6	0	Fan Tray 0 Fan 6 slot	Filled
4	2	1	0	Fan Tray 1 Fan 1 slot	Filled
4	2	2	0	Fan Tray 1 Fan 2 slot	Filled
4	2	3	0	Fan Tray 1 Fan 3 slot	Filled
4	2	4	0	Fan Tray 1 Fan 4 slot	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
4	2	5	0	Fan Tray 1 Fan 5 slot	Filled
4	2	6	0	Fan Tray 1 Fan 6 slot	Filled
4	3	1	0	Fan Tray 2 Fan 1 slot	Filled
4	3	2	0	Fan Tray 2 Fan 2 slot	Filled
4	3	3	0	Fan Tray 2 Fan 3 slot	Filled
4	3	4	0	Fan Tray 2 Fan 4 slot	Filled
4	4	1	0	Fan Tray 3 Fan 1 slot	Filled
4	4	2	0	Fan Tray 3 Fan 2 slot	Filled
4	4	3	0	Fan Tray 3 Fan 3 slot	Filled
4	4	4	0	Fan Tray 3 Fan 4 slot	Filled
4	4	5	0	Fan Tray 3 Fan 5 slot	Filled
4	4	6	0	Fan Tray 3 Fan 6 slot	Filled
7	1	0	0	FPC slot 0	Filled
7	2	0	0	FPC slot 1	Filled
7	3	0	0	FPC slot 2	Filled
7	4	0	0	FPC slot 3	Filled
7	5	0	0	FPC slot 4	Filled
7	6	0	0	FPC slot 5	Filled
7	7	0	0	FPC slot 6	Filled
7	8	0	0	FPC slot 7	Filled
7	9	0	0	FPC slot 8	Filled
7	10	0	0	FPC slot 9	Filled
7	11	0	0	FPC slot 10	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router *(continued)*

Container Index	L1	L2	L3	Description	State
7	12	0	0	FPC slot 11	Filled
7	13	0	0	FPC slot 12	Filled
7	14	0	0	FPC slot 13	Filled
7	15	0	0	FPC slot 14	Filled
7	16	0	0	FPC slot 15	Filled
7	17	0	0	FPC slot 16	Filled
7	18	0	0	FPC slot 17	Filled
7	19	0	0	FPC slot 18	Filled
7	20	0	0	FPC slot 19	Filled
8	1	1	0	PIC slot @ 0/0/*	Filled
8	1	2	0	PIC slot @ 0/1/*	Filled
8	1	3	0	PIC slot @ 0/2/*	Filled
8	1	4	0	PIC slot @ 0/3/*	Filled
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/*	Filled
8	3	1	0	PIC slot @ 2/0/*	Filled
8	3	2	0	PIC slot @ 2/1/*	Filled
8	3	3	0	PIC slot @ 2/2/*	Filled
8	3	4	0	PIC slot @ 2/3/*	Filled
8	4	1	0	PIC slot @ 3/0/*	Filled
8	4	2	0	PIC slot @ 3/1/*	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
8	4	3	0	PIC slot @ 3/2/*	Filled
8	4	4	0	PIC slot @ 3/3/*	Filled
8	5	1	0	PIC slot @ 4/0/*	Filled
8	5	2	0	PIC slot @ 4/1/*	Filled
8	5	3	0	PIC slot @ 4/2/*	Filled
8	5	4	0	PIC slot @ 4/3/*	Filled
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/*	Filled
8	7	1	0	PIC slot @ 6/0/*	Filled
8	7	2	0	PIC slot @ 6/1/*	Filled
8	7	3	0	PIC slot @ 6/2/*	Filled
8	7	4	0	PIC slot @ 6/3/*	Filled
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
8	9	1	0	PIC slot @ 8/0/*	Filled
8	9	2	0	PIC slot @ 8/1/*	Filled
8	9	3	0	PIC slot @ 8/2/*	Filled
8	9	4	0	PIC slot @ 8/3/*	Filled
8	10	1	0	PIC slot @ 9/0/*	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router *(continued)*

Container Index	L1	L2	L3	Description	State
8	10	2	0	PIC slot @ 9/1/*	Filled
8	10	3	0	PIC slot @ 9/2/*	Filled
8	10	4	0	PIC slot @ 9/3/*	Filled
8	11	1	0	PIC slot @ 10/0/*	Filled
8	11	2	0	PIC slot @ 10/1/*	Filled
8	11	3	0	PIC slot @ 10/2/*	Filled
8	11	4	0	PIC slot @ 10/3/*	Filled
8	12	1	0	PIC slot @ 11/0/*	Filled
8	12	2	0	PIC slot @ 11/1/*	Filled
8	12	3	0	PIC slot @ 11/2/*	Filled
8	12	4	0	PIC slot @ 11/3/*	Filled
8	13	1	0	PIC slot @ 12/0/*	Filled
8	13	2	0	PIC slot @ 12/1/*	Filled
8	13	3	0	PIC slot @ 12/2/*	Filled
8	13	4	0	PIC slot @ 12/3/*	Filled
8	14	1	0	PIC slot @ 13/0/*	Filled
8	14	2	0	PIC slot @ 13/1/*	Filled
8	14	3	0	PIC slot @ 13/2/*	Filled
8	14	4	0	PIC slot @ 13/3/*	Filled
8	15	1	0	PIC slot @ 14/0/*	Filled
8	15	2	0	PIC slot @ 14/1/*	Filled
8	15	3	0	PIC slot @ 14/2/*	Filled
8	15	4	0	PIC slot @ 14/3/*	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (*continued*)

Container Index	L1	L2	L3	Description	State
8	16	1	0	PIC slot @ 15/0/*	Filled
8	16	2	0	PIC slot @ 15/1/*	Empty
8	16	3	0	PIC slot @ 15/2/*	Filled
8	16	4	0	PIC slot @ 15/3/*	Empty
8	17	1	0	PIC slot @ 16/0/*	Filled
8	17	2	0	PIC slot @ 16/1/*	Filled
8	17	3	0	PIC slot @ 16/2/*	Filled
8	17	4	0	PIC slot @ 16/3/*	Filled
8	18	1	0	PIC slot @ 17/0/*	Filled
8	18	2	0	PIC slot @ 17/1/*	Filled
8	18	3	0	PIC slot @ 17/2/*	Filled
8	18	4	0	PIC slot @ 17/3/*	Filled
8	19	1	0	PIC slot @ 18/0/*	Filled
8	19	2	0	PIC slot @ 18/1/*	Filled
8	19	3	0	PIC slot @ 18/2/*	Filled
8	19	4	0	PIC slot @ 18/3/*	Filled
8	20	1	0	PIC slot @ 19/0/*	Filled
8	20	2	0	PIC slot @ 19/1/*	Filled
8	20	3	0	PIC slot @ 19/2/*	Filled
8	20	4	0	PIC slot @ 19/3/*	Filled
9	1	0	0	Routing Engine 0 slot	Filled
9	2	0	0	Routing Engine 1 slot	Filled
10	1	1	0	FPM Board slot	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SFB slot 0	Filled
15	2	0	0	SFB slot 1	Filled
15	3	0	0	SFB slot 2	Filled
15	4	0	0	SFB slot 3	Filled
15	5	0	0	SFB slot 4	Filled
15	6	0	0	SFB slot 5	Filled
15	7	0	0	SFB slot 6	Filled
15	8	0	0	SFB slot 7	Filled
20	1	1	0	MIC slot @ 0/0/*	Empty
20	1	2	0	MIC slot @ 0/1/*	Empty
20	2	1	0	MIC slot @ 1/0/*	Empty
20	2	2	0	MIC slot @ 1/1/*	Empty
20	3	1	0	MIC slot @ 2/0/*	Empty
20	3	2	0	MIC slot @ 2/1/*	Empty
20	4	1	0	MIC slot @ 3/0/*	Empty
20	4	2	0	MIC slot @ 3/1/*	Empty
20	5	1	0	MIC slot @ 4/0/*	Empty
20	5	2	0	MIC slot @ 4/1/*	Empty
20	6	1	0	MIC slot @ 5/0/*	Empty

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
20	6	2	0	MIC slot @ 5/1/*	Empty
20	7	1	0	MIC slot @ 6/0/*	Empty
20	7	2	0	MIC slot @ 6/1/*	Empty
20	8	1	0	MIC slot @ 7/0/*	Empty
20	8	2	0	MIC slot @ 7/1/*	Empty
20	9	1	0	MIC slot @ 8/0/*	Empty
20	9	2	0	MIC slot @ 8/1/*	Empty
20	10	1	0	MIC slot @ 9/0/*	Empty
20	10	2	0	MIC slot @ 9/1/*	Empty
20	11	1	0	MIC slot @ 10/0/*	Empty
20	11	2	0	MIC slot @ 10/1/*	Empty
20	12	1	0	MIC slot @ 11/0/*	Empty
20	12	2	0	MIC slot @ 11/1/*	Empty
20	13	1	0	MIC slot @ 12/0/*	Empty
20	13	2	0	MIC slot @ 12/1/*	Empty
20	14	1	0	MIC slot @ 13/0/*	Empty
20	14	2	0	MIC slot @ 13/1/*	Empty
20	15	1	0	MIC slot @ 14/0/*	Empty
20	15	2	0	MIC slot @ 14/1/*	Empty
20	16	1	0	MIC slot @ 15/0/*	Filled
20	16	2	0	MIC slot @ 15/1/*	Filled
20	17	1	0	MIC slot @ 16/0/*	Empty
20	17	2	0	MIC slot @ 16/1/*	Empty

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
20	18	1	0	MIC slot @ 17/0/*	Empty
20	18	2	0	MIC slot @ 17/1/*	Empty
20	19	1	0	MIC slot @ 18/0/*	Empty
20	19	2	0	MIC slot @ 18/1/*	Empty
20	20	1	0	MIC slot @ 19/0/*	Empty
20	20	2	0	MIC slot @ 19/1/*	Empty
21	1	0	0	PDM slot 0	Filled
21	2	0	0	PDM slot 1	Empty
21	3	0	0	PDM slot 2	Empty
21	4	0	0	PDM slot 3	Filled
22	1	0	0	PSM slot 0	Empty
22	2	0	0	PSM slot 1	Empty
22	3	0	0	PSM slot 2	Filled
22	4	0	0	PSM slot 3	Filled
22	5	0	0	PSM slot 4	Filled
22	6	0	0	PSM slot 5	Filled
22	7	0	0	PSM slot 6	Filled
22	8	0	0	PSM slot 7	Filled
22	9	0	0	PSM slot 8	Filled
22	10	0	0	PSM slot 9	Empty
22	11	0	0	PSM slot 10	Empty
22	12	0	0	PSM slot 11	Filled
22	13	0	0	PSM slot 12	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router *(continued)*

Container Index	L1	L2	L3	Description	State
22	14	0	0	PSM slot 13	Filled
22	15	0	0	PSM slot 14	Filled
22	16	0	0	PSM slot 15	Filled
22	17	0	0	PSM slot 16	Filled
22	18	0	0	PSM slot 17	Filled
23	1	0	0	ADC slot 0	Filled
23	2	0	0	ADC slot 1	Filled
23	3	0	0	ADC slot 2	Filled
23	4	0	0	ADC slot 3	Filled
23	5	0	0	ADC slot 4	Filled
23	6	0	0	ADC slot 5	Filled
23	7	0	0	ADC slot 6	Filled
23	8	0	0	ADC slot 7	Filled
23	9	0	0	ADC slot 8	Filled
23	10	0	0	ADC slot 9	Filled
23	11	0	0	ADC slot 10	Filled
23	12	0	0	ADC slot 11	Filled
23	13	0	0	ADC slot 12	Filled
23	14	0	0	ADC slot 13	Filled
23	15	0	0	ADC slot 14	Filled
23	16	0	0	ADC slot 15	Filled
23	17	0	0	ADC slot 16	Filled
23	18	0	0	ADC slot 17	Filled

Table 67: jnxFilledEntry Objects in the jnxFilledTable of an MX2020 Router (continued)

Container Index	L1	L2	L3	Description	State
23	19	0	0	ADC slot 18	Filled
23	20	0	0	ADC slot 19	Filled

[Table 68 on page 203](#) provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T640 router.

Table 68: 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
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

Table 68: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 68: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 68: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router (*continued*)

Container Index	L1	L2	L3	Description	State
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	3	0	PIC slot @ 4/2/*	Empty
8	5	4	0	PIC slot @ 4/3/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Filled
8	6	2	0	PIC slot @ 5/1/*	Filled
8	6	3	0	PIC slot @ 5/2/*	Filled
8	6	4	0	PIC slot @ 5/3/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	7	3	0	PIC slot @ 6/2/*	Empty
8	7	4	0	PIC slot @ 6/3/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Filled
8	8	2	0	PIC slot @ 7/1/*	Filled
8	8	3	0	PIC slot @ 7/2/*	Filled
8	8	4	0	PIC slot @ 7/3/*	Filled
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
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

Table 68: jnxFilledEntry Objects in the jnxFilledTable of a T640 Router (continued)

Container Index	L1	L2	L3	Description	State
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 69 on page 207](#) provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T320 router.

Table 69: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Filled
2	2	0	0	PEM slot 1	Empty
4	1	1	0	Top left front fan slot	Filled
4	1	2	0	Top left middle fan slot	Filled
4	1	3	0	Top left rear fan slot	Filled
4	1	4	0	Top right front fan slot	Filled
4	1	5	0	Top right middle fan slot	Filled
4	1	6	0	Top right rear fan slot	Filled
4	2	1	0	Bottom left front fan slot	Filled
4	2	2	0	Bottom left middle fan slot	Filled

Table 69: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 69: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 69: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router (continued)

Container Index	L1	L2	L3	Description	State
8	6	1	0	PIC slot @ 5/0/*	Empty
8	6	2	0	PIC slot @ 5/1/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Empty
8	8	2	0	PIC slot @ 7/1/*	Empty
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled

Table 70 on page 211 provides an example of **jnxFilledEntry** objects in the **jnxFilledTable** of a T4000 router.

Table 70: 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
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

Table 70: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router (continued)

Container Index	L1	L2	L3	Description	State
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	FPC slot 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
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

Table 70: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 70: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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

Table 70: jnxFilledEntry Objects in the jnxFilledTable of a T4000 Router (continued)

Container Index	L1	L2	L3	Description	State
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
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 (°C), whose object identifier is **{jnxOperatingEntry 7}**.
- **jnxOperatingCPU**—The CPU utilization percentage of the entry, whose object identifier is **{jnxOperatingEntry 8}**. It is valid for the Control Board, the FPC, and the Routing Engine. It is a 5-second rolling-weighted average calculated every second for each of the CPUs. The value is sent to the Routing Engine every 10 seconds. The value for the Routing Engine is 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}**.
- **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

- **jnxOperatingBufferCP**—The buffer pool utilization of the entry, in percentage, in the control plane, whose object identifier is **{jnxOperatingEntry 23}**.
- **jnxOperatingMemoryCP**—The entry's installed memory size, in megabytes (MB), in the control plane, whose object identifier is **{jnxOperatingEntry 24}**.

Table 71 on page 218 through Table 75 on page 244 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**
- BufferCP—**jnxOperatingBufferCP**
- Heap—**jnxOperatingHeap**
- UpTime—**jnxOperatingUpTime**
- Last Restart—**jnxOperatingLastRestart**
- Memory—**jnxOperatingMemory**
- MemoryCP—**jnxOperatingMemoryCP**

Table 71 on page 218 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of an M20 router.

Table 71: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	26	0	0	0	0	0	0	0:0:00:00.0	0
2	1	0	0	Power supply A	Running	28	0	0	0	0	0	0	0:0:00:00.0	0
2	2	0	0	Power supply B	Running	29	0	0	0	0	0	0	0:0:00:00.0	0

Table 71: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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 Operational Mode Commands*.

[Table 72 on page 219](#) provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of an MX2010 router.

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	22636600	0:0:00:00.0	0
4	1	1	0	Fan Tray 0 Fan 1	RunningAtFullSpeed	0	0	0	0	0	0	55558	0:0:00:00.00	0
4	1	2	0	Fan Tray 0 Fan 2	RunningAtFullSpeed	0	0	0	0	0	0	55558	0:0:00:00.00	0
4	1	3	0	Fan Tray 0 Fan 3	RunningAtFullSpeed	0	0	0	0	0	0	55558	0:0:00:00.00	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (*continued*)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	1	4	0	Fan Tray 0 Fan 4	RunningAtFullSpeed	0	0	0	0	0	0	55558	00000000	0
4	1	5	0	Fan Tray 0 Fan 5	RunningAtFullSpeed	0	0	0	0	0	0	55558	00000000	0
4	1	6	0	Fan Tray 0 Fan 6	RunningAtFullSpeed	0	0	0	0	0	0	55558	00000000	0
4	2	1	0	Fan Tray 1 Fan 1	RunningAtFullSpeed	0	0	0	0	0	0	55554	00000000	0
4	2	2	0	Fan Tray 1 Fan 2	RunningAtFullSpeed	0	0	0	0	0	0	55555	00000000	0
4	2	3	0	Fan Tray 1 Fan 3	RunningAtFullSpeed	0	0	0	0	0	0	55555	00000000	0
4	2	4	0	Fan Tray 1 Fan 4	RunningAtFullSpeed	0	0	0	0	0	0	55555	00000000	0
4	2	5	0	Fan Tray 1 Fan 5	RunningAtFullSpeed	0	0	0	0	0	0	55555	00000000	0
4	2	6	0	Fan Tray 1 Fan 6	RunningAtFullSpeed	0	0	0	0	0	0	55555	00000000	0
4	4	1	0	Fan Tray 3 Fan 1	RunningAtFullSpeed	0	0	0	0	0	0	55551	00000000	0
4	4	2	0	Fan Tray 3 Fan 2	RunningAtFullSpeed	0	0	0	0	0	0	55551	00000000	0
4	4	3	0	Fan Tray 3 Fan 3	RunningAtFullSpeed	0	0	0	0	0	0	55552	00000000	0
4	4	4	0	Fan Tray 3 Fan 4	RunningAtFullSpeed	0	0	0	0	0	0	55552	00000000	0
4	4	5	0	Fan Tray 3 Fan 5	RunningAtFullSpeed	0	0	0	0	0	0	55552	00000000	0
4	4	6	0	Fan Tray 3 Fan 6	RunningAtFullSpeed	0	0	0	0	0	0	55552	00000000	0
7	1	0	0	FPC: MPC 3D 16x 10GE @ 0/*/*	Running	68	10	0	0	13	18	39221	2:256:66.11	2048

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (*continued*)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	2	0	0	FPC: MPC Type 3 @ 1/*/*	Running	67	8	0	0	13	14	38114	22:56:77:18	2048
7	3	0	0	FPC: MPC Type 3 @ 2/*/*	Running	67	8	0	0	13	14	37361	22:56:84:71	2048
7	4	0	0	FPC: MPC 3D 16x 10GE @ 3/*/*	Running	67	10	0	0	13	18	36647	22:56:91:86	2048
7	5	0	0	FPC: MPC Type 3 @ 4/*/*	Running	68	8	0	0	13	14	35537	22:57:02:96	2048
7	6	0	0	FPC: MPC Type 4-1 @ 5/*/*	Running	68	13	0	0	14	16	23725	22:58:21:08	2048
7	7	0	0	FPC: MPC 3D 16x 10GE @ 6/*/*	Running	71	10	0	0	13	18	34151	22:57:16:82	2048
7	8	0	0	FPC: MPC 3D 16x 10GE @ 7/*/*	Running	72	12	0	0	13	18	33385	22:57:24:48	2048
7	9	0	0	FPC: MPC Type 3 @ 8/*/*	Running	70	8	0	0	13	14	32342	22:57:34:92	2048
7	10	0	0	FPC: MPCE Type 2 3D @ 9/*/*	Running	69	9	0	0	13	11	31552	22:57:42:82	2048
8	1	1	0	PIC: 4x 10GE(LAN) SFP+ @ 0/0/*	Running	0	0	0	0	0	0	20939	22:58:48:95	0
8	1	2	0	PIC: 4x 10GE(LAN) SFP+ @ 0/1/*	Running	0	0	0	0	0	0	20924	22:58:49:11	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	1	3	0	PIC: 4x 10GE(LAN) SFP+ @ 0/2/*	Running	0	0	0	0	0	0	20906	22:58:4927	0
8	1	4	0	PIC: 4x 10GE(LAN) SFP+ @ 0/3/*	Running	0	0	0	0	0	0	20892	22:58:4943	0
8	2	1	0	PIC: 1X100GE CFP @ 1/0/*	Running	0	0	0	0	0	0	22874	22:58:2960	0
8	2	3	0	PIC: 1X100GE CFP @ 1/2/*	Running	0	0	0	0	0	0	22648	22:58:3187	0
8	3	1	0	PIC: 2X40GE QSFP @ 2/0/*	Running	0	0	0	0	0	0	23413	22:58:2421	0
8	3	3	0	PIC: 2X40GE QSFP @ 2/2/*	Running	0	0	0	0	0	0	22570	22:58:3265	0
8	4	1	0	PIC: 4x 10GE(LAN) SFP+ @ 3/0/*	Running	0	0	0	0	0	0	18561	22:58:7274	0
8	4	2	0	PIC: 4x 10GE(LAN) SFP+ @ 3/1/*	Running	0	0	0	0	0	0	18545	22:58:7291	0
8	4	3	0	PIC: 4x 10GE(LAN) SFP+ @ 3/2/*	Running	0	0	0	0	0	0	18529	22:58:7307	0
8	4	4	0	PIC: 4x 10GE(LAN) SFP+ @ 3/3/*	Running	0	0	0	0	0	0	18512	22:58:7323	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (*continued*)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	5	1	0	PIC: 2X40GE QSFP @ 4/0/*	Running	0	0	0	0	0	0	21692	22:58:41.44	0
8	5	3	0	PIC: 2X40GE QSFP @ 4/2/*	Running	0	0	0	0	0	0	20848	22:58:49.88	0
8	6	1	0	PIC: 8X10GE SFPP @ 5/0/*	Running	0	0	0	0	0	0	4872	22:60:09.65	0
8	6	2	0	PIC: 8X10GE SFPP @ 5/1/*	Running	0	0	0	0	0	0	4084	22:60:17.53	0
8	6	3	0	PIC: 8X10GE SFPP @ 5/2/*	Running	0	0	0	0	0	0	4063	22:60:17.74	0
8	6	4	0	PIC: 8X10GE SFPP @ 5/3/*	Running	0	0	0	0	0	0	3986	22:60:18.51	0
8	7	1	0	PIC: 4x 10GE(LAN) SFP+ @ 6/0/*	Running	0	0	0	0	0	0	16044	22:58:97.93	0
8	7	2	0	PIC: 4x 10GE(LAN) SFP+ @ 6/1/*	Running	0	0	0	0	0	0	16028	22:58:98.09	0
8	7	3	0	PIC: 4x 10GE(LAN) SFP+ @ 6/2/*	Running	0	0	0	0	0	0	16011	22:58:98.26	0
8	7	4	0	PIC: 4x 10GE(LAN) SFP+ @ 6/3/*	Running	0	0	0	0	0	0	15997	22:58:98.41	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	8	1	0	PIC: 4x 10GE(LAN) SFP+ @ 7/0/*	Running	0	0	0	0	0	0	15302	2259:05:35	0
8	8	2	0	PIC: 4x 10GE(LAN) SFP+ @ 7/1/*	Running	0	0	0	0	0	0	15287	2259:05:51	0
8	8	3	0	PIC: 4x 10GE(LAN) SFP+ @ 7/2/*	Running	0	0	0	0	0	0	15271	2259:05:83	0
8	8	4	0	PIC: 4x 10GE(LAN) SFP+ @ 7/3/*	Running	0	0	0	0	0	0	15255	2259:05:83	0
8	9	1	0	PIC: 2X40GE QSFP @ 8/0/*	Running	0	0	0	0	0	0	18416	22:58:74.21	0
8	9	3	0	PIC: 2X40GE QSFP @ 8/2/*	Running	0	0	0	0	0	0	17574	22:58:82.64	0
8	10	1	0	PIC: 1x10GE XFP @ 9/0/*	Running	0	0	0	0	0	0	21450	22:58:30.89	0
8	10	2	0	PIC: 1x10GE XFP @ 9/1/*	Running	0	0	0	0	0	0	21130	22:58:47.09	0
8	10	3	0	PIC: 1x10GE XFP @ 9/2/*	Running	0	0	0	0	0	0	20952	22:58:48.87	0
8	10	4	0	PIC: 1x10GE XFP @ 9/3/*	Running	0	0	0	0	0	0	20652	22:58:51.87	0
9	1	0	0	Routing Engine 0	Running	66	29	0	80530668	12	0	22636600	0:0:0:0.0	17152
9	2	0	0	Routing Engine 1	Standby	72	0	0	80530668	11	0	22557000	48820	17152

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
10	1	1	0	FPM Board	Running	0	0	0	0	0	0	55387	0	0
12	1	0	0	CB 0	Running	69	0	0	0	0	0	55491	0	0
12	2	0	0	CB1	Standby	70	0	0	0	0	0	0	0	0
14	1	0	0	SPMB 0	Running	63	83	0	0	22	2	53137	22:55:27.03	2816
14	2	0	0	SPMB 1	Standby	67	0	0	0	22	0	0	22:55:29.10	2816
15	1	0	0	SFB 0	Running	70	0	0	0	0	0	22636820	00000000	0
15	2	0	0	SFB 1	Running	69	0	0	0	0	0	22636821	00000000	0
15	3	0	0	SFB 2	Running	68	0	0	0	0	0	22636821	00000000	0
15	4	0	0	SFB 3	Running	70	0	0	0	0	0	22636821	00000000	0
15	5	0	0	SFB 4	Running	71	0	0	0	0	0	22636821	00000000	0
15	6	0	0	SFB 5	Running	72	0	0	0	0	0	22636822	00000000	0
15	7	0	0	SFB 6	Running	71	0	0	0	0	0	22636822	00000000	0
15	8	0	0	SFB 7	Running	72	0	0	0	0	0	22636822	00000000	0
20	2	1	0	MIC: 1X100GE CFP @ 1/0/*	Running	0	0	0	0	0	0	22882	22:58:29.61	0
20	2	2	0	MIC: 1X100GE CFP @ 1/1/*	Running	0	0	0	0	0	0	22656	22:58:31.87	0
20	3	1	0	MIC: 2X40GE QSFP @ 2/0/*	Running	0	0	0	0	0	0	23420	22:58:24.22	0
20	3	2	0	MIC: 2X40GE QSFP @ 2/1/*	Running	0	0	0	0	0	0	22577	22:58:32.66	0
20	5	1	0	MIC: 2X40GE QSFP @ 4/0/*	Running	0	0	0	0	0	0	21699	22:58:41.45	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
20	5	2	0	MIC: 2X40GE QSFP @ 4/1/*	Running	0	0	0	0	0	0	20855	22:58:49.89	0
20	9	1	0	MIC: 2X40GE QSFP @ 8/0/*	Running	0	0	0	0	0	0	18421	22:58:74.22	0
20	9	2	0	MIC: 2X40GE QSFP @ 8/1/*	Running	0	0	0	0	0	0	17579	22:58:82.65	0
20	10	1	0	MIC: 3D 2x 10GE XFP @ 9/0/*	Running	0	0	0	0	0	0	21135	22:58:47.09	0
20	10	2	0	MIC: 3D 2x 10GE XFP @ 9/1/*	Running	0	0	0	0	0	0	20657	22:58:51.88	0
21	1	0	0	PDM 0	Running	0	0	0	0	0	0	55574	00:00:00.00	0
22	1	0	0	PSM 0	Running	71	0	0	0	0	0	55576	00:00:00.00	0
22	2	0	0	PSM 1	Running	69	0	0	0	0	0	55576	00:00:00.00	0
22	3	0	0	PSM 2	Running	69	0	0	0	0	0	55576	00:00:00.00	0
22	4	0	0	PSM 3	Running	69	0	0	0	0	0	55576	00:00:00.00	0
22	5	0	0	PSM 4	Running	69	0	0	0	0	0	55576	00:00:00.00	0
22	6	0	0	PSM 5	Running	69	0	0	0	0	0	55576	00:00:00.00	0
22	7	0	0	PSM 6	Running	68	0	0	0	0	0	55576	00:00:00.00	0
22	8	0	0	PSM 7	Running	67	0	0	0	0	0	55576	00:00:00.00	0
22	9	0	0	PSM 8	Running	65	0	0	0	0	0	55576	00:00:00.00	0
23	1	0	0	ADC 0	Running	73	0	0	0	0	0	22636827	00:00:00.00	0
23	2	0	0	ADC 1	Running	74	0	0	0	0	0	22636827	00:00:00.00	0
23	3	0	0	ADC 2	Running	73	0	0	0	0	0	22636827	00:00:00.00	0

Table 72: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
23	4	0	0	ADC 3	Running	73	0	0	0	0	0	22636827	00000000	0
23	5	0	0	ADC 4	Running	74	0	0	0	0	0	22636828	00000000	0
23	6	0	0	ADC 5	Running	73	0	0	0	0	0	22636828	00000000	0
23	7	0	0	ADC 6	Running	74	0	0	0	0	0	22636828	00000000	0
23	8	0	0	ADC 7	Running	75	0	0	0	0	0	22636828	00000000	0
23	9	0	0	ADC 8	Running	75	0	0	0	0	0	22636829	00000000	0
23	10	0	0	ADC 9	Running	75	0	0	0	0	0	22636829	00000000	0

Table 73 on page 227 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of an MX2020 router.

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 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	4665800	0:0:0:0:0:0	0
4	1	1	0	Fan Tray 0 Fan 1	Running	0	0	0	0	0	0	14988767	0:0:0:0:0:0	0
4	1	2	0	Fan Tray 0 Fan 2	Running	0	0	0	0	0	0	14988767	0:0:0:0:0:0	0
4	1	3	0	Fan Tray 0 Fan 3	Running	0	0	0	0	0	0	14988768	0:0:0:0:0:0	0
4	1	4	0	Fan Tray 0 Fan 4	Running	0	0	0	0	0	0	14988768	0:0:0:0:0:0	0
4	1	5	0	Fan Tray 0 Fan 5	Running	0	0	0	0	0	0	14988768	0:0:0:0:0:0	0
4	1	6	0	Fan Tray 0 Fan 6	Running	0	0	0	0	0	0	14988768	0:0:0:0:0:0	0
4	2	1	0	Fan Tray 1 Fan 1	Running	0	0	0	0	0	0	14988764	0:0:0:0:0:0	0
4	2	2	0	Fan Tray 1 Fan 2	Running	0	0	0	0	0	0	14988764	0:0:0:0:0:0	0
4	2	3	0	Fan Tray 1 Fan 3	Running	0	0	0	0	0	0	14988765	0:0:0:0:0:0	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (continued)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	2	4	0	Fan Tray 1 Fan 4	Running	0	0	0	0	0	0	14988765	0:0:00:00:0	0
4	2	5	0	Fan Tray 1 Fan 5	Running	0	0	0	0	0	0	14988765	0:0:00:00:0	0
4	2	6	04	Fan Tray 1 Fan 6	Running	0	0	0	0	0	0	14988765	0:0:00:00:0	0
4	3	1	0	Fan Tray 2 Fan 1	Running	0	0	0	0	0	0	14988761	0:0:00:00:0	0
4	3	2	0	Fan Tray 2 Fan 2	Running	0	0	0	0	0	0	14988761	0:0:00:00:0	0
4	3	3	0	Fan Tray 2 Fan 3	Running	0	0	0	0	0	0	14988762	0:0:00:00:0	0
4	3	4	0	Fan Tray 2 Fan 4	Running	0	0	0	0	0	0	14988762	0:0:00:00:0	0
4	3	5	0	Fan Tray 2 Fan 5	Running	0	0	0	0	0	0	14988762	0:0:00:00:0	0
4	3	6	0	Fan Tray 2 Fan 6	Running	0	0	0	0	0	0	14988762	0:0:00:00:0	0
4	4	1	0	Fan Tray 3 Fan 1	Running	0	0	0	0	0	0	14988758	0:0:00:00:0	0
4	4	2	0	Fan Tray 3 Fan 2	Running	0	0	0	0	0	0	14988758	0:0:00:00:0	0
4	4	3	0	Fan Tray 3 Fan 3	Running	0	0	0	0	0	0	14988759	0:0:00:00:0	0
4	4	4	0	Fan Tray 3 Fan 4	Running	0	0	0	0	0	0	14988759	0:0:00:00:0	0
4	4	5	0	Fan Tray 3 Fan 5	Running	0	0	0	0	0	0	14988759	0:0:00:00:0	0
4	4	6	0	Fan Tray 3 Fan 6	Running	0	0	0	0	0	0	14988759	0:0:00:00:0	0
7	1	0	0	FPC: MPC 3D 16x 10GE @ 0/*/*	Running	43	12	0	0	13	18	14970350	3:3:68:18:91	2048

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	2	0	0	FPC: MPC 3D 16x 10GE @ 1/*/*	Running	42	9	0	0	13	18	14969114	3:3:68:31:28	2048
7	3	0	0	FPC: MPC 3D 16x 10GE @ 2/*/*	Running	41	9	0	0	13	18	14968402	3:3:68:38:39	2048
7	4	0	0	FPC: MPC 3D 16x 10GE @ 3/*/*	Running	42	9	0	0	13	18	14967777	3:3:68:44:65	2048
7	5	0	0	FPC: MPC 3D 16x 10GE @ 4/*/*	Running	42	9	0	0	13	18	14966743	3:3:68:55:00	2048
7	6	0	0	FPC: MPC 3D 16x 10GE @ 5/*/*	Running	42	9	0	0	13	18	14966065	3:3:68:61:78	2048
7	7	0	0	FPC: MPC 3D 16x 10GE @ 6/*/*	Running	42	10	0	0	13	18	14965240	3:3:68:70:03	2048
7	8	0	0	FPC: MPC 3D 16x 10GE @ 7/*/*	Running	43	10	0	0	13	18	14964346	3:3:68:78:97	2048
7	9	0	0	FPC: MPC 3D 16x 10GE @ 8/*/*	Running	42	10	0	0	13	18	14963598	3:3:68:86:46	2048
7	10	0	0	FPC: MPC 3D 16x 10GE @ 9/*/*	Running	43	11	0	0	13	18	14962745	3:3:68:94:99	2048
7	11	0	0	FPC: MPC 3D 16x 10GE @ 10/*/*	Running	44	11	0	0	13	18		3:3:69:02:19	2048

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
7	12	0	0	FPC: MPC 3D 16x 10GE @ 11/*/*	Running	41	8	0	0	13	18	14961167	3:3:69:10:77	2048
7	13	0	0	FPC: MPC 3D 16x 10GE @ 12/*/*	Running	40	10	0	0	13	18	14960311	3:3:69:19:33	2048
7	14	0	0	FPC: MPC 3D 16x 10GE @ 13/*/*	Running	40	10	0	0	13	18	14959198	3:3:69:30:47	2048
7	15	0	0	FPC: MPC 3D 16x 10GE @ 14/*/*	Running	41	10	0	0	13	18	14958385	3:3:69:38:60	2048
7	16	0	0	FPC: MPC 3D 16x 10GE @ 15/*/*	Running	41	10	0	0	13	18	14957402	3:3:69:48:43	2048
7	17	0	0	FPC: MPC 3D 16x 10GE @ 16/*/*	Running	41	8	0	0	13	18	14956721	3:3:69:55:25	2048
7	18	0	0	FPC: MPC 3D 16x 10GE @ 17/*/*	Running	40	10	0	0	13	18	14955764	3:3:69:64:82	2048
7	19	0	0	FPC: MPC 3D 16x 10GE @ 18/*/*	Running	41	9	0	0	13	18	14954818	3:3:69:74:29	2048
7	20	0	0	FPC: MPC 3D 16x 10GE @ 19/*/*	Running	42	10	0	0	13	18	14954091	3:3:69:81:56	2048
8	1	1	08	PIC: 4x 10GE(LAN) SFP+ @ 0/0/*	Running	0	0	0	0	0	0	14952090	3:3:70:01:57	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	1	2	0	PIC: 4x 10GE(LAN) SFP+ @ 0/1/*	Running	0	0	0	0	0	0	14952074	3:3:70:01:74	0
8	1	3	0	PIC: 4x 10GE(LAN) SFP+ @ 0/2/*	Running	0	0	0	0	0	0	14952058	3:3:70:01:90	0
8	1	4	0	PIC: 4x 10GE(LAN) SFP+ @ 0/3/*	Running	0	0	0	0	0	0	14952042	3:3:70:02:05	0
8	2	1	0	PIC: 4x 10GE(LAN) SFP+ @ 1/0/*	Running	0	0	0	0	0	0	14949459	3:3:70:27:89	0
8	2	2	0	PIC: 4x 10GE(LAN) SFP+ @ 1/1/*	Running	0	0	0	0	0	0	14949443	3:3:70:28:05	0
8	2	3	0	PIC: 4x 10GE(LAN) SFP+ @ 1/2/*	Running	0	0	0	0	0	0	14949409	3:3:70:28:39	0
8	2	4	0	PIC: 4x 10GE(LAN) SFP+ @ 1/3/*	Running	0	0	0	0	0	0	14949405	3:3:70:28:42	0
8	3	1	0	PIC: 4x 10GE(LAN) SFP+ @ 2/0/*	Running	0	0	0	0	0	0	14950344	3:3:70:19:05	0
8	3	2	0	PIC: 4x 10GE(LAN) SFP+ @ 2/1/*	Running	0	0	0	0	0	0	14950333	3:3:70:19:16	0
8	3	3	0	PIC: 4x 10GE(LAN) SFP+ @ 2/2/*	Running	0	0	0	0	0	0	14950299	3:3:70:19:50	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	3	4	0	PIC: 4x 10GE(LAN) SFP+ @ 2/3/*	Running	0	0	0	0	0	0	14950297	3:3:70:19:52	0
8	4	1	0	PIC: 4x 10GE(LAN) SFP+ @ 3/0/*	Running	0	0	0	0	0	0	14949322	3:3:70:29:28	0
8	4	2	0	PIC: 4x 10GE(LAN) SFP+ @ 3/1/*	Running	0	0	0	0	0	0	14949279	3:3:70:29:71	0
8	4	3	0	PIC: 4x 10GE(LAN) SFP+ @ 3/2/*	Running	0	0	0	0	0	0	14949277	3:3:70:29:73	0
8	4	4	0	PIC: 4x 10GE(LAN) SFP+ @ 3/3/*	Running	0	0	0	0	0	0	14949274	3:3:70:29:76	0
8	5	1	0	PIC: 4x 10GE(LAN) SFP+ @ 4/0/*	Running	0	0	0	0	0	0	14948483	3:3:70:37:68	0
8	5	2	0	PIC: 4x 10GE(LAN) SFP+ @ 4/1/*	Running	0	0	0	0	0	0	14948466	3:3:70:37:85	0
8	5	3	0	PIC: 4x 10GE(LAN) SFP+ @ 4/2/*	Running	0	0	0	0	0	0	14948450	3:3:70:38:01	0
8	5	4	0	PIC: 4x 10GE(LAN) SFP+ @ 4/3/*	Running	0	0	0	0	0	0	14948435	3:3:70:38:16	0
8	6	1	0	PIC: 4x 10GE(LAN) SFP+ @ 5/0/*	Running	0	0	0	0	0	0	14947756	3:3:70:44:95	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	6	2	0	PIC: 4x 10GE(LAN) SFP+ @ 5/1/*	Running	0	0	0	0	0	0	14947740	3:3:70:45:12	0
8	6	3	0	PIC: 4x 10GE(LAN) SFP+ @ 5/2/*	Running	0	0	0	0	0	0	14947724	3:3:70:45:28	0
8	6	4	0	PIC: 4x 10GE(LAN) SFP+ @ 5/3/*	Running	0	0	0	0	0	0	14947618	3:3:70:46:33	0
8	7	1	0	PIC: 4x 10GE(LAN) SFP+ @ 6/0/*	Running	0	0	0	0	0	0	14945107	3:3:70:71:45	0
8	7	2	0	PIC: 4x 10GE(LAN) SFP+ @ 6/1/*	Running	0	0	0	0	0	0	14945066	3:3:70:71:87	0
8	7	3	0	PIC: 4x 10GE(LAN) SFP+ @ 6/2/*	Running	0	0	0	0	0	0	14945032	3:3:70:72:21	0
8	7	4	0	PIC: 4x 10GE(LAN) SFP+ @ 6/3/*	Running	0	0	0	0	0	0	14944994	3:3:70:72:60	0
8	8	1	0	PIC: 4x 10GE(LAN) SFP+ @ 7/0/*	Running	0	0	0	0	0	0	14944474	3:3:70:77:80	0
8	8	2	0	PIC: 4x 10GE(LAN) SFP+ @ 7/1/*	Running	0	0	0	0	0	0	14944438	3:3:70:78:16	0
8	8	3	0	PIC: 4x 10GE(LAN) SFP+ @ 7/2/*	Running	0	0	0	0	0	0	14944400	3:3:70:78:54	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (continued)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	8	4	0	PIC: 4x 10GE(LAN) SFP+ @ 7/3/*	Running	0	0	0	0	0	0	14943595	3:3:70:78:59	0
8	9	1	0	PIC: 4x 10GE(LAN) SFP+ @ 8/0/*	Running	0	0	0	0	0	0	14943595	3:3:70:86:60	0
8	9	2	0	PIC: 4x 10GE(LAN) SFP+ @ 8/1/*	Running	0	0	0	0	0	0	14943547	3:3:70:87:08	0
8	9	3	0	PIC: 4x 10GE(LAN) SFP+ @ 8/2/*	Running	0	0	0	0	0	0	14943541	3:3:70:87:14	0
8	9	4	0	PIC: 4x 10GE(LAN) SFP+ @ 8/3/*	Running	0	0	0	0	0	0	14943537	3:3:70:87:18	0
8	10	1	0	PIC: 4x 10GE(LAN) SFP+ @ 9/0/*	Running	0	0	0	0	0	0	14943136	3:3:70:91:19	0
8	10	2	0	PIC: 4x 10GE(LAN) SFP+ @ 9/1/*	Running	0	0	0	0	0	0	14943098	3:3:70:91:57	0
8	10	3	0	PIC: 4x 10GE(LAN) SFP+ @ 9/2/*	Running	0	0	0	0	0	0	14943052	3:3:70:92:04	0
8	10	4	0	PIC: 4x 10GE(LAN) SFP+ @ 9/3/*	Running	0	0	0	0	0	0	14943024	3:3:70:92:32	0
8	11	1	0	PIC: 4x 10GE(LAN) SFP+ @ 10/0/*	Running	0	0	0	0	0	0	14942186	3:3:71:00:70	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	11	2	0	PIC: 4x 10GE(LAN) SFP+ @ 10/1/*	Running	0	0	0	0	0	0	14942151	3:3:71:01:05	0
8	11	3	0	PIC: 4x 10GE(LAN) SFP+ @ 10/2/*	Running	0	0	0	0	0	0	14942076	3:3:71:01:81	0
8	11	4	0	PIC: 4x 10GE(LAN) SFP+ @ 10/3/*	Running	0	0	0	0	0	0	14942032	3:3:71:02:25	0
8	12	1	0	PIC: 4x 10GE(LAN) SFP+ @ 11/0/*	Running	0	0	0	0	0	0	14942436	3:3:70:98:21	0
8	12	2	0	PIC: 4x 10GE(LAN) SFP+ @ 11/1/*	Running	0	0	0	0	0	0	14942385	3:3:70:98:72	0
8	12	3	0	PIC: 4x 10GE(LAN) SFP+ @ 11/2/*	Running	0	0	0	0	0	0	14942337	3:3:70:99:20	0
8	12	4	0	PIC: 4x 10GE(LAN) SFP+ @ 11/3/*	Running	0	0	0	0	0	0	14942291	3:3:70:99:67	0
8	13	1	0	PIC: 4x 10GE(LAN) SFP+ @ 12/0/*	Running	0	0	0	0	0	0	14940160	3:3:71:20:98	0
8	13	2	0	PIC: 4x 10GE(LAN) SFP+ @ 12/1/*	Running	0	0	0	0	0	0	14940119	3:3:71:21:38	0
8	13	3	0	PIC: 4x 10GE(LAN) SFP+ @ 12/2/*	Running	0	0	0	0	0	0	14939995	3:3:71:22:63	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (continued)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	13	4	0	PIC: 4x 10GE(LAN) SFP+ @ 12/3/*	Running	0	0	0	0	0	0	14939951	3:3:71:23:07	0
8	14	1	0	PIC: 4x 10GE(LAN) SFP+ @ 13/0/*	Running	0	0	0	0	0	0	14940967	3:3:71:12:92	0
8	14	2	0	PIC: 4x 10GE(LAN) SFP+ @ 13/1/*	Running	0	0	0	0	0	0	14940927	3:3:71:13:32	0
8	14	3	0	PIC: 4x 10GE(LAN) SFP+ @ 13/2/*	Running	0	0	0	0	0	0	14940887	3:3:71:13:72	0
8	14	4	0	PIC: 4x 10GE(LAN) SFP+ @ 13/3/*	Running	0	0	0	0	0	0	14940848	3:3:71:14:11	0
8	15	1	0	PIC: 4x 10GE(LAN) SFP+ @ 14/0/*	Running	0	0	0	0	0	0	14938064	3:3:71:41:95	0
8	15	2	0	PIC: 4x 10GE(LAN) SFP+ @ 14/1/*	Running	0	0	0	0	0	0	14937989	3:3:71:42:70	0
8	15	3	0	PIC: 4x 10GE(LAN) SFP+ @ 14/2/*	Running	0	0	0	0	0	0	14937940	3:3:71:43:20	0
8	15	4	0	PIC: 4x 10GE(LAN) SFP+ @ 14/3/*	Running	0	0	0	0	0	0	14937899	3:3:71:43:61	0
8	16	1	0	PIC: 4x 10GE(LAN) SFP+ @ 15/0/*	Running	0	0	0	0	0	0	14937190	3:3:71:50:71	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	16	2	0	PIC: 4x 10GE(LAN) SFP+ @ 15/1/*	Running	0	0	0	0	0	0	14937095	3:3:71:51:66	0
8	16	3	0	PIC: 4x 10GE(LAN) SFP+ @ 15/2/*	Running	0	0	0	0	0	0	14937043	3:3:71:52:18	0
8	16	4	0	PIC: 4x 10GE(LAN) SFP+ @ 15/3/*	Running	0	0	0	0	0	0	14936968	3:3:71:52:93	0
8	17	1	0	PIC: 4x 10GE(LAN) SFP+ @ 16/0/*	Running	0	0	0	0	0	0	14938250	3:3:71:40:11	0
8	17	2	0	PIC: 4x 10GE(LAN) SFP+ @ 16/1/*	Running	0	0	0	0	0	0	14938205	3:3:71:40:55	0
8	17	3	0	PIC: 4x 10GE(LAN) SFP+ @ 16/2/*	Running	0	0	0	0	0	0	14938156	3:3:71:41:06	0
8	17	4	0	PIC: 4x 10GE(LAN) SFP+ @ 16/3/*	Running	0	0	0	0	0	0	14938108	3:3:71:41:53	0
8	18	1	0	PIC: 4x 10GE(LAN) SFP+ @ 17/0/*	Running	0	0	0	0	0	0	14935631	3:3:71:66:31	0
8	18	2	0	PIC: 4x 10GE(LAN) SFP+ @ 17/1/*	Running	0	0	0	0	0	0	14935589	3:3:71:66:74	0
8	18	3	0	PIC: 4x 10GE(LAN) SFP+ @ 17/2/*	Running	0	0	0	0	0	0	14935546	3:3:71:67:17	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (continued)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
8	18	4	0	PIC: 4x 10GE(LAN) SFP+ @ 17/3/*	Running	0	0	0	0	0	0	14935451	3:3:71:68:12	0
8	19	1	0	PIC: 4x 10GE(LAN) SFP+ @ 18/0/*	Running	0	0	0	0	0	0	14936276	3:3:71:59:87	0
8	19	2	0	PIC: 4x 10GE(LAN) SFP+ @ 18/1/*	Running	0	0	0	0	0	0	14936231	3:3:71:60:32	0
8	19	3	0	PIC: 4x 10GE(LAN) SFP+ @ 18/2/*	Running	0	0	0	0	0	0	14936177	3:3:71:60:86	0
8	19	4	0	PIC: 4x 10GE(LAN) SFP+ @ 18/3/*	Running	0	0	0	0	0	0	14936136	3:3:71:61:28	0
8	20	1	0	PIC: 4x 10GE(LAN) SFP+ @ 19/0/*	Running	0	0	0	0	0	0	14934058	3:3:71:82:06	0
8	20	2	0	PIC: 4x 10GE(LAN) SFP+ @ 19/1/*	Running	0	0	0	0	0	0	14934014	3:3:71:82:49	0
8	20	3	0	PIC: 4x 10GE(LAN) SFP+ @ 19/2/*	Running	0	0	0	0	0	0	14933940	3:3:71:83:24	0
8	20	4	0	PIC: 4x 10GE(LAN) SFP+ @ 19/3/*	Running	0	0	0	0	0	0	14933886	3:3:71:83:78	0
9	1	0	0	Routing Engine 0	Running	37	74	2	805306368	14	0	48659800	0	17152
9	2	0	0	Routing Engine 1	Standby	37	0	0	805306368	11	0	48660000	0	17152
10	1	1	0	FPM Board	Running	0	0	0	0	0	0	14988508	0	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
12	1	0	0	CB 0	Running	44	0	0	0	0	0	14988692	0	0
12	2	0	0	CB 1	Standby	43	0	0	0	0	0	0	0	0
14	1	0	0	SPMB 0	Running	35	15	0	0	22	3	14985877	33666389	2816
14	2	0	0	SPMB 1	Standby	35	0	0	0	22	0	0	33666202	2816
15	1	0	0	SFB 0	Running	45	0	0	0	0	0	48660333	0	0
15	2	0	0	SFB 1	Running	44	0	0	0	0	0	48660333	0	0
15	3	0	0	SFB 2	Running	44	0	0	0	0	0	48660333	0	0
15	4	0	0	SFB 3	Running	45	0	0	0	0	0	48660334	0	0
15	5	0	0	SFB 4	Running	46	0	0	0	0	0	48660334	0	0
15	6	0	0	SFB 5	Running	46	0	0	0	0	0	48660334	0	0
15	7	0	0	SFB 6	Running	46	0	0	0	0	0	48660335	0	0
15	8	0	0	SFB 7	Running	45	0	0	0	0	0	48660335	0	0
21	1	0	0	PDM 0	Running	0	0	0	0	0	0	14988820	0	0
21	2	0	0	PDM 1	Running	0	0	0	0	0	0	14988821	0	0
21	3	0	0	PDM 2	Running	0	0	0	0	0	0	14988821	0	0
21	4	0	0	PDM 3	Running	0	0	0	0	0	0	14988821	0	0
22	1	0	0	PSM 0	Running	37	0	0	0	0	0	14988825	0	0
22	2	0	0	PSM 1	Running	36	0	0	0	0	0	14988825	0	0
22	3	0	0	PSM 2	Running	36	0	0	0	0	0	14988825	0	0
22	4	0	0	PSM 3	Running	36	0	0	0	0	0	14988825	0	0
22	5	0	0	PSM 4	Running	36	0	0	0	0	0	14988825	0	0
22	8	0	0	PSM 7	Running	38	0	0	0	0	0	14988826	0	0
22	9	0	0	PSM 8	Running	36	0	0	0	0	0	14988826	0	0
22	10	0	0	PSM 9	Running	39	0	0	0	0	0	14988826	0	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (continued)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
22	11	0	0	PSM 10	Running	37	0	0	0	0	0	14988826	0	0
22	12	0	0	PSM 11	Running	38	0	0	0	0	0	14988826	0	0
22	13	0	0	PSM 12	Running	38	0	0	0	0	0	14988826	0	0
22	14	0	0	PSM 13	Running	37	0	0	0	0	0	14988827	0	0
22	15	0	0	PSM 14	Running	37	0	0	0	0	0	14988827	0	0
22	16	0	0	PSM 15	Running	38	0	0	0	0	0	14988827	0	0
22	17	0	0	PSM 16	Running	38	0	0	0	0	0	14988827	0	0
22	18	0	0	PSM 17	Running	38	0	0	0	0	0	14988827	0	0
23	1	0	0	ADC 0	Running	49	0	0	0	0	0	48660364	0	0
23	2	0	0	ADC 1	Running	49	0	0	0	0	0	48660364	0	0
23	3	0	0	ADC 2	Running	48	0	0	0	0	0	48660405	0	0
23	4	0	0	ADC 3	Running	50	0	0	0	0	0	48660405	0	0
23	5	0	0	ADC 4	Running	48	0	0	0	0	0	48660405	0	0
23	6	0	0	ADC 5	Running	49	0	0	0	0	0	48660405	0	0
23	7	0	0	ADC 6	Running	48	0	0	0	0	0	48660407	0	0
23	8	0	0	ADC 7	Running	52	0	0	0	0	0	48660407	0	0
23	9	0	0	ADC 8	Running	50	0	0	0	0	0	48660405	0	0
23	10	0	0	ADC 9	Running	50	0	0	0	0	0	48660408	0	0
23	11	0	0	ADC 10	Running	49	0	0	0	0	0	48660408	0	0
23	12	0	0	ADC 11	Running	52						48660408	0	0
23	13	0	0	ADC 12	Running	51	0	0	0	0	0	48660409	0	0
23	14	0	0	ADC 13	Running	55	0	0	0	0	0	48660409	0	0
23	15	0	0	ADC 14	Running	54	0	0	0	0	0	48660409	0	0
23	16	0	0	ADC 15	Running	55	0	0	0	0	0	48660410	0	0

Table 73: jnxOperatingEntry Objects in the jnxOperatingTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
23	17	0	0	ADC 16	Running	54	0	0	0	0	0	48660410	0	0
23	18	0	0	ADC 17	Running	53	0	0	0	0	0	48660410	0	0
23	19	0	0	ADC 18	Running	52	0	0	0	0	0	48660411	0	0
23	20	0	0	ADC 19	Running	51	0	0	0	0	0	48660411	0	0

To verify the size of the memory, use the **show chassis fpc** and **show chassis routing-engine** commands. For more information about the output of these commands, see the *Junos OS Operational Mode Commands*.

Table 74 on page 241 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of a T640 router.

Table 74: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	—	—	—	—	—	—	—	—
2	2	0	0	PEM 1	Running	29	—	—	—	—	—	—	—	—
4	1	1	0	Top left front fan	Running	0	—	—	—	—	—	—	—	—
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

Table 74: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom right rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	2	0	Bottom blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	3	0	Middle blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	4	0	Top blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	5	0	Second blower from top	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	Running	0	1	0	512	41	3	138367	0:18:56:48.81	512
7	2	1	0	FPC @ 1/0/* top 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

Table 74: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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
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 75 on page 244 provides an example of `jnxOperatingEntry` objects in the `jnxOperatingTable` of a T320 router.

Table 75: `jnxOperatingEntry` Objects in the `jnxOperatingTable` of a T320 Router

Index	L1	L2	L3	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
2	1	0	0	PEM 0	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	1	0	Top left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	2	0	Top left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	3	0	Top left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	4	0	Top right front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	5	0	Top right middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Top right rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	1	0	Bottom left front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	2	0	Bottom left middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	3	0	Bottom left rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
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

Table 75: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router (continued)

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

Table 75: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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 76 on page 246 provides an example of **jnxOperatingEntry** objects in the **jnxOperatingTable** of a T4000 router.

Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	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

Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router (continued)

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

Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router (continued)

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

Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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	2924 2924	0	0:18:56:52.85	0
8	5	2	0	PIC: 12x10GE (LAN/WAN) SFPP @ 4/1/*	Running	33	0	0	0	0	2924 2924	0	0:18:56:52.85	0
9	1	0	0	Routing Engine 0	Running	35	3	0	8506 8506	20	0	2494800	0:19:20:30.07	8950
9	2	0	0	Routing Engine 1	Standby	32	0	0	8506 8506	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

Table 76: jnxOperatingEntry Objects in the jnxOperatingTable of a T4000 Router (continued)

Index	L1	L2	B	Description	State	Temp	CPU	SR	DRAM	Buffer	Heap	UpTime	LastRestart	Memory
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 77 on page 252 through Table 81 on page 256 provide examples of `jnxRedundancyEntry` objects. The following column headings for each table are abbreviated to correspond to the parts of the `jnxOperatingTable` objects:

- Contents index—`jnxRedundancyContentsIndex`
- L1—`jnxRedundancyL1Index`
- L2—`jnxRedundancyL2Index`
- L3—`jnxRedundancyL3Index`
- Description—`jnxRedundancyDescr`
- Config—`jnxRedundancyConfig`
- State—`jnxRedundancyState`
- Count—`jnxRedundancySwitchoverCount`
- Time—`jnxRedundancySwitchoverTime`
- Reason—`jnxRedundancySwitchoverReason`
- Heartbeat—`jnxKeepaliveHeartbeat`
- Timeout—`jnxRedundancyKeepaliveTimeout`
- Elapsed—`jnxRedundancyKeepaliveElapsed`
- Loss—`jnxRedundancyKeepaliveLoss`

Table 77 on page 252 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of an M20 router.

Table 77: 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	SSB1	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 78 on page 253 provides an example of **jnxRedundancyEntry** objects in the **jnxRedundancyTable** of an MX2010 router.

Table 78: jnxRedundancyEntry Objects in the jnxRedundancyTable of an MX2010 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	1	0	User switched	0	360	70	0
9	2	0	0	Routing Engine 1	Present	Present	0	0	Other	0	0	0	0

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an MX2010 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
CPU temperature               33 degrees C / 91 degrees F
DRAM                         3313 MB
Memory utilization            25 percent
CPU utilization:
  User                       0 percent
  Background                  0 percent
  Kernel                      1 percent
  Interrupt                   1 percent
  Idle                        98 percent
Model                        RE-S-1800x4
Serial ID                     9009094134
Start time                   2012-07-18 17:15:00 PDT
Uptime                       25 days, 7 hours, 9 minutes, 45 seconds

```

```

Last reboot reason          Router rebooted after a normal shutdown.
Load averages:              1 minute   5 minute   15 minute
                             0.00       0.00       0.00

Routing Engine status:
Slot 1:
Current state               Present

```

Table 79 on page 254 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of an MX2020 router.

Table 79: jnxRedundancyEntry Objects in the jnxRedundancyTable of an MX2020 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	User switched	0	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 MX2020 router is as follows:

```

user@host> show chassis routing-engine
Routing Engine status:
Slot 0:
Current state                Master
Election priority            Master (default)
Temperature                  39 degrees C / 102 degrees F
CPU temperature              36 degrees C / 96 degrees F
DRAM                        3313 MB
Memory utilization           24 percent
CPU utilization:
  User                       0 percent
  Background                 0 percent
  Kernel                     2 percent
  Interrupt                   1 percent
  Idle                       97 percent
Model                       RE-S-1800x4
Serial ID                    9009089721
Start time                   2012-08-09 15:32:30 PDT
Uptime                       3 days, 9 hours, 1 minute, 11 seconds
Last reboot reason          Router rebooted after a normal shutdown.
Load averages:               1 minute   5 minute   15 minute
                             0.02       0.02       0.00

Routing Engine status:
Slot 1:
Current state                Backup
Election priority            Backup (default)
Temperature                  45 degrees C / 113 degrees F
CPU temperature              41 degrees C / 105 degrees F
DRAM                        17152 MB
Memory utilization           12 percent
CPU utilization:
  User                       0 percent
  Background                 0 percent

```

```

Kernel          0 percent
Interrupt       0 percent
Idle           100 percent
Model          RE-S-1800x4
Serial ID      9009089729
Start time     2012-08-08 11:22:04 PDT
Uptime         4 days, 13 hours, 11 minutes, 23 seconds
Last reboot reason Router rebooted after a normal shutdown.

```

Table 80 on page 255 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of a T640 router.

Table 80: 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
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 81 on page 256](#) provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of a T320 router.

Table 81: 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
  Start time 2002-03-27 15:25:07 PST
  Uptime 26 days, 22 hours, 21 minutes, 44 seconds

```

Table 82 on page 257 provides an example of `jnxRedundancyEntry` objects in the `jnxRedundancyTable` of a T4000 router.

Table 82: 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)
 - switchInterfaceBoard(9)
 - processorMezzanineBoardForSIB(10)
 - portInterfaceCard(11)
 - craftInterfacePanel(12)
 - fan(13)
 - forwardingEngineBoard(13)
 - protectedSystemDomain(13)
 - lineCardChassis(14)
 - powerDistributionModule(17)
 - powerSupplyModule(18)
 - switchFabricBoard(19)
 - adapterCard(20)
- **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.



NOTE: For information about using the **jnxFruSlot** object for a member router in an MX Series Virtual Chassis configuration, see *Virtual Chassis Slot Number Mapping for Use with SNMP*.

- **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 83 on page 261 through Table 90 on page 324 provide examples of **jnxFruEntry** objects. The following column headings for each table are abbreviated to correspond to the parts of the **jnxFruEntry** objects:

- Contents index—**jnxFruContentsIndex**
- L1—**jnxFruL1Index**
- L2—**jnxFruL2Index**
- L3—**jnxFruL3Index**
- Name—**jnxFruName**
- Type—**jnxFruType**
- Slot—**jnxFruSlot**
- State—**jnxFruState**
- Temp—**jnxFruTemp**
- Offline—**jnxFruOffline**
- PowerOff—**jnxFruPowerOff**
- PowerOn—**jnxFruPowerOn**
- Uptime—**jnxFruPowerUpTime**

Table 83 on page 261 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M10 router.

Table 83: 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	0:0:0:0:0:0	0:0:11:08.73	264319
2	2	0	0	Power supply B	powerEntryModule	2	Empty	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
4	1	1	0	Left fan 1	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
4	1	2	0	Left fan 2	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
4	1	3	0	Left fan 3	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
4	1	4	0	Left fan 4	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0

Table 83: jnxFruContents Objects in the jnxFruTable of an M10 Router (continued)

Index	L1	L2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
6	1	0	0	FEB Internet Processor II	controlBoard	1	Online	24	None	0:00:00.00	0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	Online	24	None	0:00:00.00	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	Online	24	None	0:00:00.00	0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	Ready	24	None	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: 1x Monitor @ 0/1/*	portInterfaceCard	1	Ready	24	None	0:00:00.00	0:00:00.00	0
8	1	3	0	PIC: 1x OC-12 ATM, MM @ 0/2/*	portInterfaceCard	1	Ready	24	None	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: 4x T3 @ 0/3/*	portInterfaceCard	1	Ready	24	None	0:00:00.00	0:00:00.00	0
8	2	1	0	PIC: 4x OC-3 SONET, SMIR @ 1/0/*	portInterfaceCard	2	Ready	24	None	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	Ready	24	None	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	Ready	24	None	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: 2x OC-3 ATM, MM @ 1/3/*	portInterfaceCard	2	Ready	24	None	0:00:00.00	0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	Online	27	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 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
  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 84 on page 264 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M20 router.

Table 84: JnxFruContents Objects in the jnxFruTable of an M20 Router

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntryModule	1	Empty	0	None	0:00:00.00	0:00:00.00	0
2	2	0	0	Power supply B	powerEntryModule	2	Online	25	None	0:00:00.00	0:00:43.45	24993357
4	1	0	0	Rear fan	fan	1	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	0	0	Front upper fan	fan	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	3	0	0	Front middle fan	fan	3	Present	0	None	0:00:00.00	0:00:00.00	0
4	4	0	0	Front bottom fan	fan	4	Present	0	None	0:00:00.00	0:00:00.00	0
6	1	0	0	SSB 0	controlBoard	1	Present	0	None	0:00:00.00	0:00:00.00	0
6	2	0	0	SSB 1 Internet Processor I	controlBoard	2	Online	29	None	0:00:00.00	0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexible PicConcentrator	1	Empty	0	None	0:00:00.00	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexible PicConcentrator-	2	Online	27	None	0:00:00.00	0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexible PicConcentrator	3	Empty	0	None	0:00:00.00	0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexible PicConcentrator	4	Online	27	None	0:00:00.00	0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	Offline	0	None	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	Offline	28	None	0:00:00.00	0:00:00.00	0

Table 84: JnxFruContents Objects in the jnxFruTable of an M20 Router (continued)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	Offline	0	None	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	Offline	0	None	0:00:00.00	0:00:00.00	0
8	2	1	0	PIC: 1x Tunnel @ 1/0/*	portInterfaceCard	2	Ready	0	None	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: 4x T3 @ 1/1/*	portInterfaceCard	2	Ready	0	None	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	Ready	27	None	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: 1x G/E, 1000Base-SX @ 1/3/*	portInterfaceCard	2	Ready	27	None	0:00:00.00	0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	Offline	27	None	0:00:00.00	0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	Offline	0	None	0:00:00.00	0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	Offline	0	None	0:00:00.00	0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	Offline	0	None	0:00:00.00	0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	Ready	0	None	0:00:00.00	0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	Ready	28	None	0:00:00.00	0:00:00.00	0
8	4	3	0	PIC: 2x OC-3 SONET, SMIR @ 3/2/*	portInterfaceCard	4	Ready	28	None	0:00:00.00	0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	Ready	28	None	0:00:00.00	0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	Online	25	None	0:00:00.00	0:00:00.00	0

Table 84: JnxFruContents Objects in the jnxFruTable of an M20 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
```

Slot	State	Temp (C)	CPU Utilization (%)	Memory Utilization (%)
			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
  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 85 on page 268 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M160 router.

Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router

Index	I	L	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
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

Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)

Index	I	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
6	3	2	0	SFM 2 SPR Internet Processor II	switchingAnd ForwardingMode	3	Online	44	None	0:00:03.20	0:00:00.00	0
6	4	1	0	SFM 3 SPP	switchingAnd ForwardingMode	4	Offline	0	Config- ured to power off	0:00:03.22	0:00:00.00	0
6	4	2	0	SFM 3 SPR	switchingAnd ForwardingMode	4	Offline	0	Config- ured to power off	0:00:03.22	0:00:00.00	0
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 SMLR @ 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

Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)

Index	I	2	B	Name	Type	St	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	Online	40	None	0:00:00.00	0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	portInterfaceCard	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	portInterfaceCard	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	2	Online	46	None	0:00:00.00	0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	Offline	0	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

Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)

Index	I	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	4	1	0	PIC: 1x OC-192 SMLR @ 3/0/*	portInterfaceCard	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	2	0	PIC continued	portInterfaceCard	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	3	0	PIC continued	portInterfaceCard	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	4	4	0	PIC continued	portInterfaceCard	4	Offline	0	Error	0:13:08.12	0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	Online	30	None	0:00:02.32	0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	Online	41	None	0:00:02.34	0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard (11)	7	Empty	0	None	0:00:00.00	0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard (11)	7	Empty	0	None	0:00:00.00	0:00:00.00	0

Table 85: jnxFruContents Objects in the jnxFruTable of an M160 Router (continued)

Index	I	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
8	8	3	0	PIC: 1x OC-12 SONET, MM @ 7/2/*	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
Midplane             REV 02   710-001245   AB4113
FPM CMB              REV 01   710-001642   AA9721
FPM Display          REV 01   710-001647   AA2995
CIP                  REV 02   710-001593   AA9886
PEM 0                Rev 01   740-001243   KJ35782        DC
PEM 1                Rev 01   740-001243   kj35756        DC
PCG 0                REV 01   710-001568   AA9796
PCG 1                REV 01   710-001568   AA9895
Routing Engine 0     REV01    740-003239   AARCH00        RE-2.0
Routing Engine 1
MCS 0                REV 03   710-001226   AA9779
SFM 0 SPP            REV 07   710-001228   AE5504
SFM 0 SPR            REV 03   710-002189   AE4707        Internet Processor II
SFM 2 SPP            REV 06   710-001228   AB3133
SFM 2 SPR            REV 01   710-002189   AB2941        Internet Processor II
SFM 3 SPP            REV 07   710-001228   AV3167
SFM 3 SPR            REV 04   710-002189   AV3439        Internet Processor II
FPC 0                REV 02   710-001611   AA9518        FPC Type 2
CPU                  REV 02   710-001217   AA9572
FPC 1                REV 03   710-001255   AA9812        FPC Type 1
CPU
FPC 2                REV 02   710-001611   AA9527        FPC Type 2
CPU                  REV 02   710-001217   AA9592
FPC 3                REV 01   710-003061   HB2029        FPC Type 0C192
CPU                  REV 05   710-001217   AF5950
PIC 0                REV 01   750-003063   HB2029        1x 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 86 on page 274](#) provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M40 router.

Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power supply A	powerEntry Module	1	Online	0	None	0:0:0:0:0:0	0:0:0:0:0:0	101974
2	2	0	0	Power supply B	powerEntry Module	2	Empty	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
3	1	0	0	Top impeller	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
3	2	0	0	Bottom impeller	fan	2	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
4	1	0	0	Rear left fan	fan	1	Present	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0

Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router (continued)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	2	0	0	Rear center fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear right fan	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
5	1	0	0	Host controller	routingEngine	1	Online	37	None	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SCB Internet Processor I	controlBoard	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePic Concentrator	1	Online	28	None	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexible PicConcentrator	2	Online	29	None	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexible PicConcentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePic Concentrator	4	Online	24	None	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePic Concentrator	5	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePic Concentrator	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	7	0	0	FPC: 1x OC-48 SONET, SMIR @ 6/*/*	flexiblePic Concentrator	7	Online	28	None	0:0:00:00.00	0:0:00:00.00	0
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, 100Base-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

Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router (*continued*)

Index	I	2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	3	0	PIC: 4x TI, RJ48 @ 0/2/*	portInterfaceCard	1	Ready	24	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	1	4	0	PIC: 1x COC12, SMIR @ 0/3/*	portInterfaceCard	1	Ready	24	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	2	1	0	PIC: 2x OC-3 ATM, MM @ 1/0/*	portInterfaceCard	2	Ready	27	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	Ready	27	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	2	3	0	PIC: 2x T3 @ 1/2/*	portInterfaceCard	2	Ready	27	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	2	4	0	PIC: 1x CSTM1, SMIR @ 1/3/*	portInterfaceCard	2	Ready	27	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	Offline	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	Offline	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	Offline	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	Offline	0	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	Ready	24	None	0:0:0:0:0:0	0:0:0:0:0:0	0
8	4	2	0	PIC: 4x F/E, 100 BASE-TX @ 3/1/*	portInterfaceCard	4	Ready	24	None	0:0:0:0:0:0	0:0:0:0:0:0	0

Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router (*continued*)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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-MDSO @ 3/3/*	portInterfaceCard	4	Ready	24	None	0:0:00:00:00	0:0:00:00:00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	Ready	27	None	0:0:00:00:00	0:0:00:00:00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	Ready	27	None	0:0:00:00:00	0:0:00:00:00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	Ready	27	None	0:0:00:00:00	0:0:00:00:00	0
8	5	4	0	PIC: @ 4/3/*	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

Table 86: jnxFruContents Objects in the jnxFruTable of an M40 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	Offline	0	None	0:0:00:00:00	0:0:00:00:00	0
8	8	3	0	PIC: @ 7/2/*	portInterfaceCard	8	Offline	0	None	0:0:00:00:00	0:0:00:00:00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	Offline	0	None	0:0:00:00:00	0:0:00:00:00	0
9	1	0	0	Routing Engine	routingEngine	1	Online	0	None	0:0:00:00:00	0:0:00:00:00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an M40 router is 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
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 87 on page 280 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an M40e router.

Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	Present	0	None	0:00:00.00	0:00:25.99	208927
2	2		0	PEM 1	powerEntryModule	2	Online	0	None	0:00:00.00	0:00:25.99	208928
4	1	0	0	Front top blower	fan	1	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	1	0	Fan tray front left	fan	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	2	0	Fan tray front right	fan	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	3	0	Fan tray rear left	fan	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	2	4	0	Fan tray rear right	fan	2	Present	0	None	0:00:00.00	0:00:00.00	0
4	3	0	0	Rear top blower	fan	3	Present		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	switchingAndForwardingModule	1	Empty	0	None	0:00:00.00	0:00:00.00	0
6	1	2	0	SFM 0 SPR	switchingAndForwardingModule	1	Empty	0	None	0:00:00.00	0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAndForwardingModule	2	Online	42	None	0:00:21.95	0:00:00.00	0
6	2	2	0	SFM 1 SPR Internet Processor II	switchingAndForwardingModule	2	Online	42	None	0:00:21.95	0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	Online	41	None	0:00:21.85	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	Empty	0	None	0:00:00.00	0:00:00.00	0

Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router (continued)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router (continued)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	1	0	PIC: 1x OC-12 SONET, MM @ 2/0/*	portInterfaceCard	3	Online	41	None	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: 1x OC-12 SONET, MM @ 2/1/*	portInterfaceCard	3	Online	41	None	0:0:00:21.85	0:0:00:00.00	0
8	3	3	0	PIC: 1x OC-12 SONET, MM @ 2/2/*	portInterfaceCard	3	Online	41	—	0:0:00:21.85	0:0:00:00.00	—
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	Online	41	—	0:0:00:21.85	0:0:00:00.00	—
8	4	1	0	PIC: 1x OC-48 SONET, SMIR @ 3/0/*	portInterfaceCard	4	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterfaceCard	4	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
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

Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router (continued)

Index	I1	I2	I3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	Online	38	—	0:0:00:21.89	0:0:00:00.00	0
8	6	3	0	PIC: 1x OC-12 SONET, SMIR @ 5/2/*	portInterfaceCard	6	Online	38	—	0:0:00:21.89	0:0:00:00.00	0
8	6	4	0	PIC: 1x OC-12 SONET, MM @ 5/3/*	portInterfaceCard	6	Online	38	—	0:0:00:21.89	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard	7	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard	7	Empty	0	—	0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: 8x FE-FX, 100 BASE-FX @ 7/0/*	portInterfaceCard	8	Online	46	—	0:0:00:21.91	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	Online	46	—	0:0:00:21.91	0:0:00:00.00	0
8	8	3	0	PIC: 1x Link Service(4) @ 7/2/*	portInterfaceCard	8	Online	46	—	0:0:00:21.91	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	1	Online	46	—	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	2	Online	46	—	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	1	Present	34	—	0:0:00:00.00	0:0:00:00.00	0

Table 87: jnxFruContents Objects in the jnxFruTable of an M40e Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
10	1	1	0	FPM CMB	frontPanelModule	1	Online	28	—	0:0:0:0:0:0	0:0:0:0:0:0	0
10	1	2	0	FPM Display	frontPanelModule	1	Online	28	—	0:0:0:0:0:0	0:0:0:0:0:0	0
11	1	0	0	PCG 0	clockGenerator	1	Online	45	—	0:0:0:0:0:0	0:0:0:0:0:0	0
11	2	0	0	PCG 1	clockGenerator	2	Online	50	—	0:0:0:0:0:0	0:0:0:0:0:0	0
12	1	0	0	MCS 0	controlBoard	1	Online	46	—	0:0:0:0:0:0	0:0:0:0:0:0	0
12	2	0	0	MCS 1	controlBoard	2	Online	0	—	0:0:0:0:0:0	0:0:0:0:0:0	0
13	1	0	0	CIP	craftInterfacePanel	1	Present	0	—	0:0:0:0:0:0	0:0:0:0:0:0	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
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
FPC 0                REV 01    710-005078   BE0642        Internet Processor II
CPU                  REV 01    710-004600   BD2496        M40e-FPC Type 1
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:
  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 88 on page 285 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an MX2010 router.

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router

Index	L1	L2	L3	Name	Type	St	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	1	1	0	Fan Tray 0 Fan 1	fan	0	Online	0	None	000000	22041848	246978

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	St	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	1	2	0	Fan Tray 0 Fan 2	fan	0	Online	0	None	00000000	2204:1848	246978
4	1	3	0	Fan Tray 0 Fan 3	fan	0	Online	0	None	00000000	2204:1848	246978
4	1	4	0	Fan Tray 0 Fan 4	fan	0	Online	0	None	00000000	2204:1848	246978
4	1	5	0	Fan Tray 0 Fan 5	fan	0	Online	0	None	00000000	2204:1848	246978
4	1	6	0	Fan Tray 0 Fan 6	fan	0	Online	0	None	00000000	2204:1848	246978
4	2	1	0	Fan Tray 1 Fan 1	fan	1	Online	0	None	00000000	2204:1852	246973
4	2	2	0	Fan Tray 1 Fan 2	fan	1	Online	0	None	00000000	2204:1852	246973
4	2	3	0	Fan Tray 1 Fan 3	fan	1	Online	0	None	00000000	2204:1852	246973
4	2	4	0	Fan Tray 1 Fan 4	fan	1	Online	0	None	00000000	2204:1852	246974
4	2	5	0	Fan Tray 1 Fan 5	fan	1	Online	0	None	00000000	2204:1852	246974
4	2	6	0	Fan Tray 1 Fan 6	fan	1	Online	0	None	00000000	2204:1852	246974
4	4	1	0	Fan Tray 3 Fan 1	fan	3	Online	0	None	00000000	2204:1856	246970
4	4	2	0	Fan Tray 3 Fan 2	fan	3	Online	0	None	00000000	2204:1856	246970

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	St	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	4	3	0	Fan Tray 3 Fan 3	fan	3	Online	0	None	00000000	22041856	246970
4	4	4	0	Fan Tray 3 Fan 4	fan	3	Online	0	None	00000000	22041856	246970
4	4	5	0	Fan Tray 3 Fan 5	fan	3	Online	0	None	00000000	22041856	246970
4	4	6	0	Fan Tray 3 Fan 6	fan	3	Online	0	None	00000000	22041856	246971
7	1	0	0	FPC: MPC 3D 16x 10GE @ 0/*/*	flexiblePicConcentrator	0	Online	69	None	00000000	22056473	232354
7	2	0	0	FPC: MPC Type 3 @ 1/*/*	flexiblePicConcentrator	1	Online	68	None	00000000	22057330	231498
7	3	0	0	FPC: MPC Type 3 @ 2/*/*	flexiblePicConcentrator	2	Online	68	None	00000000	22058197	230630
7	4	0	0	FPC: MPC 3D 16x 10GE @ 3/*/*	flexiblePicConcentrator	3	Online	68	None	00000000	22059056	229772
7	5	0	0	FPC: MPC Type 3 @ 4/*/*	flexiblePicConcentrator	4	Online	69	None	00000000	22059972	228856
7	6	0	0	FPC: MPC Type 4-1 @ 5/*/*	flexiblePicConcentrator	5	Online	69	None	00000000	22116660	172167

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	St	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	7	0	0	FPC: MPC 3D 16x 10GE @ 6/*/*	flexiblePicConcentrator	6	Online	72	None	00000000	22061646	227183
7	8	0	0	FPC: MPC 3D 16x 10GE @ 7/*/*	flexiblePicConcentrator	7	Online	73	None	00000000	22062416	226412
7	9	0	0	FPC: MPC Type 3 @ 8/*/*	flexiblePicConcentrator	8	Online	71	None	00000000	22063072	225757
7	10	0	0	FPC: MPCE Type 2 3D @ 9/*/*	flexiblePicConcentrator	9	Online	70	None	00000000	22063989	224839
8	1	1	0	PIC: 4x 10GE(LAN) SFP+ @ 0/0/*	portInterfaceCard	0	Online	0	None	00000000	220716421	212407
8	1	2	0	PIC: 4x 10GE(LAN) SFP+ @ 0/1/*	portInterfaceCard	0	Online	0	None	00000000	220716421	212408
8	1	3	0	PIC: 4x 10GE(LAN) SFP+ @ 0/2/*	portInterfaceCard	0	Online	0	None	00000000	220716421	212408
8	1	4	0	PIC: 4x 10GE(LAN) SFP+ @ 0/3/*	portInterfaceCard	0	Online	0	None	00000000	220716421	212408
8	2	1	0	PIC: 1X100GE CFP @ 1/0/*	portInterfaceCard	1	Online	0	None	00000000	220714290	214540
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	1	Empty	0	None	00000000	00000000	0

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	3	0	PIC: 1X100GE CFP @ 1/2/*	portInterfaceCard	1	Online	0	None	00000000	22074292	214537
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	1	Empty	0	None	00000000	00000000	0
8	3	1	0	PIC: 2X40GE QSFP @ 2/0/*	portInterfaceCard	2	Online	0	None	00000000	22073793	215037
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	2	Empty	0	None	00000000	00000000	0
8	3	3	0	PIC: 2X40GE QSFP @ 2/2/*	portInterfaceCard	2	Online	0	None	00000000	22073796	215034
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	2	Empty	0	None	00000000	00000000	0
8	4	1	0	PIC: 4x 10GE(LAN) SFP+ @ 3/0/*	portInterfaceCard	3	Online	0	None	00000000	22079002	209829
8	4	2	0	PIC: 4x 10GE(LAN) SFP+ @ 3/1/*	portInterfaceCard	3	Online	0	None	00000000	22079002	209829
8	4	3	0	PIC: 4x 10GE(LAN) SFP+ @ 3/2/*	portInterfaceCard	3	Online	0	None	00000000	22079002	209829
8	4	4	0	PIC: 4x 10GE(LAN) SFP+ @ 3/3/*	portInterfaceCard	3	Online	0	None	00000000	22079002	209829
8	5	1	0	PIC: 2X40GE QSFP @ 4/0/*	portInterfaceCard	4	Online	0	None	00000000	22075411	213420
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	4	Empty	0	None	00000000	00000000	0

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	5	3	0	PIC: 2X40GE QSFP @ 4/2/*	portInterfaceCard	4	Online	0	None	00000000	22075414	213417
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	4	Empty	0	None	00000000	00000000	0
8	6	1	0	PIC: 8X10GE SFPP @ 5/0/*	portInterfaceCard	5	Online	0	None	00000000	22:14:81.71	140661
8	6	2	0	PIC: 8X10GE SFPP @ 5/1/*	portInterfaceCard	5	Online	0	None	00000000	22:14:81.71	140661
8	6	3	0	PIC: 8X10GE SFPP @ 5/2/*	portInterfaceCard	5	Online	0	None	00000000	22:14:81.71	140661
8	6	4	0	PIC: 8X10GE SFPP @ 5/3/*	portInterfaceCard	5	Online	0	None	00000000	22:14:81.71	140661
8	7	1	0	PIC: 4x 10GE(LAN) SFP+ @ 6/0/*	portInterfaceCard	6	Online	0	None	00000000	22081643	207191
8	7	2	0	PIC: 4x 10GE(LAN) SFP+ @ 6/1/*	portInterfaceCard	6	Online	0	None	00000000	22081643	207191
8	7	3	0	PIC: 4x 10GE(LAN) SFP+ @ 6/2/*	portInterfaceCard	6	Online	0	None	00000000	22081643	207191
8	7	4	0	PIC: 4x 10GE(LAN) SFP+ @ 6/3/*	portInterfaceCard	6	Online	0	None	00000000	22081643	207191
8	8	1	0	PIC: 4x 10GE(LAN) SFP+ @ 7/0/*	portInterfaceCard	7	Online	0	None	00000000	22082333	206500

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	2	0	PIC: 4x 10GE(LAN) SFP+ @ 7/1/*	portInterfaceCard	7	Online	0	None	00000000	22082333	206500
8	8	3	0	PIC: 4x 10GE(LAN) SFP+ @ 7/2/*	portInterfaceCard	7	Online	0	None	00000000	22082333	206500
8	8	4	0	PIC: 4x 10GE(LAN) SFP+ @ 7/3/*	portInterfaceCard	7	Online	0	None	00000000	22082333	206500
8	9	1	0	PIC: 2X40GE QSFP @ 8/0/*	portInterfaceCard	8	Online	0	None	00000000	22078466	210367
8	9	2	0	PIC: @ 8/1/*	portInterfaceCard	8	Empty	0	None	00000000	00000000	0
8	9	3	0	PIC: 2X40GE QSFP @ 8/2/*	portInterfaceCard	8	Online	0	None	00000000	22078469	210365
8	9	4	0	PIC: @ 8/3/*	portInterfaceCard	8	Empty	0	None	00000000	00000000	0
8	10	1	0	PIC: 1x 10GE XFP @ 9/0/*	portInterfaceCard	9	Online	0	None	00000000	22075883	212951
8	10	2	0	PIC: 1x 10GE XFP @ 9/1/*	portInterfaceCard	9	Online	0	None	00000000	22075885	212949
8	10	3	0	PIC: 1x 10GE XFP @ 9/2/*	portInterfaceCard	9	Online	0	None	00000000	22075892	212943
8	10	4	0	PIC: 1x 10GE XFP @ 9/3/*	portInterfaceCard	9	Online	0	None	00000000	22075898	212936

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
9	1	0	0	Routing Engine 0	routingEngine	0	Online	67	None	00000000	2204:1757	247077
9	2	0	0	Routing Engine 1	routingEngine	1	Online	73	None	00000000	2204:1758	247077
10	1	1	0	FPM Board	frontPanelModule	0	Online	0	None	00000000	2204:2029	246806
12	1	0	0	CB 0	controlBoard	0	Online	70	None	00000000	2204:1925	246910
12	2	0	0	CB 1	controlBoard	1	Online	72	None	00000000	2204:1940	246896
14	1	0	0	SPMB 0	processorMezzanineBoardForSIB	0	Online	63	None	22:11:53.74	2204:4324	244511
14	2	0	0	SPMB 1	processorMezzanineBoardForSIB	1	Online	69	None	00000000	2204:4092	244744
15	1	0	0	SFB 0	switchFabricBoard	0	Online	71	None	00000000	2204:6587	242249
15	2	0	0	SFB 1	switchFabricBoard	1	Online	70	None	00000000	2204:7737	241100
15	3	0	0	SFB 2	switchFabricBoard	2	Online	69	None	00000000	2204:8730	240106
15	4	0	0	SFB 3	switchFabricBoard	3	Online	71	None	00000000	2204:9802	239035
15	5	0	0	SFB 4	switchFabricBoard	4	Online	73	None	00000000	2205:0873	237963
15	6	0	0	SFB 5	switchFabricBoard	5	Online	74	None	00000000	2205:1946	236892
15	7	0	0	SFB 6	switchFabricBoard	6	Online	73	over temp Shutdown	00000000	2205:3018	235819
15	8	0	0	SFB 7	switchFabricBoard	7	Online	75	None	00000000	2205:4091	234746
20	1	1	0	MIC: @ 0/0/*	portInterfaceCard	0	Empty	0	None	00000000	00000000	0
20	1	2	0	MIC: @ 0/1/*	portInterfaceCard	0	Empty	0	None	00000000	00000000	0
20	2	1	0	MIC: 1X100GE CFP @ 1/0/*	portInterfaceCard	1	Online	0	None	00000000	2207:4498	214339

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
20	2	2	0	MIC: 1X100GE CFP @ 1/1/*	portInterfaceCard	1	Online	0	None	00000000	22074761	214077
20	3	1	0	MIC: 2X40GE QSFP @ 2/0/*	portInterfaceCard	2	Online	0	None	00000000	22074629	214209
20	3	2	0	MIC: 2X40GE QSFP @ 2/1/*	portInterfaceCard	2	Online	0	None	00000000	22075471	213367
20	4	1	0	MIC: @ 3/0/*	portInterfaceCard	3	Empty	0	None	00000000	00000000	0
20	4	2	0	MIC: @ 3/1/*	portInterfaceCard	3	Empty	0	None	00000000	00000000	0
20	5	1	0	MIC: 2X40GE QSFP @ 4/0/*	portInterfaceCard	4	Online	0	None	00000000	22075260	212579
20	5	2	0	MIC: 2X40GE QSFP @ 4/1/*	portInterfaceCard	4	Online	0	None	00000000	22077090	211748
20	6	1	0	MIC: @ 5/0/*	portInterfaceCard	5	Empty	0	None	00000000	00000000	0
20	6	2	0	MIC: @ 5/1/*	portInterfaceCard	5	Empty	0	None	00000000	00000000	0
20	7	1	0	MIC: @ 6/0/*	portInterfaceCard	6	Empty	0	None	00000000	00000000	0
20	7	2	0	MIC: @ 6/1/*	portInterfaceCard	6	Empty	0	None	00000000	00000000	0
20	8	1	0	MIC: @ 7/0/*	portInterfaceCard	7	Empty	0	None	00000000	00000000	0
20	8	2	0	MIC: @ 7/1/*	portInterfaceCard	7	Empty	0	None	00000000	00000000	0

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Stt	State	Temp	Offline	PowerOff	PowerOn	Uptime
20	9	1	0	MIC: 2X40GE QSFP @ 8/0/*	portInterfaceCard	8	Online	0	None	00000000	22079301	209539
20	9	2	0	MIC: 2X40GE QSFP @ 8/1/*	portInterfaceCard	8	Online	0	None	00000000	22080145	208696
20	10	1	0	MIC: 3D 2x10GE XFP @ 9/0/*	portInterfaceCard	9	Online	0	None	00000000	22076274	212567
20	10	2	0	MIC: 3D 2x10GE XFP @ 9/1/*	portInterfaceCard	9	Online	0	None	00000000	22076762	212079
21	1	0	0	PDM 0	powerDistributionModule	0	Online	0	None	00000000	22041848	246994
21	2	0	0	PDM 1	powerDistributionModule	1	Empty	0	None	00000000	00000000	0
22	1	0	0	PSM 0	powerSupplyModule	0	Online	73	None	00000000	22041846	246996
22	2	0	0	PSM 1	powerSupplyModule	1	Online	71	None	00000000	22041846	246996
22	3	0	0	PSM 2	powerSupplyModule	2	Online	71	None	00000000	22041846	246996
22	4	0	0	PSM 3	powerSupplyModule	3	Online	71	None	00000000	22041846	246996
22	5	0	0	PSM 4	powerSupplyModule	4	Online	71	None	00000000	22041847	246996
22	6	0	0	PSM 5	powerSupplyModule	5	Online	70	None	00000000	22041847	246996
22	7	0	0	PSM 6	powerSupplyModule	6	Online	69	None	00000000	22041847	246996
22	8	0	0	PSM 7	powerSupplyModule	7	Online	68	None	00000000	22041847	246996
22	9	0	0	PSM 8	powerSupplyModule	8	Online	66	None	00000000	22041847	246996
23	1	0	0	ADC 0	adapterCard	0	Online	75	None	00000000	22056428	232415
23	2	0	0	ADC 1	adapterCard	1	Online	75	None	00000000	22057244	231600
23	3	0	0	ADC 2	adapterCard	2	Online	74	None	00000000	22058057	230786
23	4	0	0	ADC 3	adapterCard	3	Online	75	None	00000000	22058880	229963

Table 88: jnxFruContents Objects in the jnxFruTable of an MX2010 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
23	5	0	0	ADC 4	adapterCard	4	Online	76	None	00000000	22059744	229100
23	6	0	0	ADC 5	adapterCard	5	Online	75	None	00000000	22116565	172279
23	7	0	0	ADC 6	adapterCard	6	Online	76	None	00000000	22061377	227467
23	8	0	0	ADC 7	adapterCard	7	Online	77	None	00000000	22062200	226644
23	9	0	0	ADC 8	adapterCard	8	Online	78	None	00000000	22063037	225807
23	10	0	0	ADC 9	adapterCard	9	Online	77	None	00000000	22063883	224962

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from an MX2010 router is as follows:

```

user@host> show chassis hardware
Hardware inventory:
Item              Version  Part number  Serial number  Description
Chassis              REV 26   750-044636   JN11E233DAFK   MX2010
Midplane             REV 01   711-044557   ABAB9357       Lower Backplane
Midplane 1           REV 04   711-032426   ACAJ1677       Upper Backplane
PMP                   REV 08   760-044634   ABBV9726       Power Midplane
FPM Board             REV 01   740-045050   1E02224000P    Front Panel Display
PSM 0                 REV 01   740-045050   1E02224000M    DC 52V Power Supply
Module
PSM 1                 REV 01   740-045050   1E022240010    DC 52V Power Supply
Module
PSM 2                 REV 01   740-045050   1E022240010    DC 52V Power Supply
Module
PSM 3                 REV 01   740-045050   1E02224000G    DC 52V Power Supply
Module
PSM 4                 REV 01   740-045050   1E022240013    DC 52V Power Supply
Module
PSM 5                 REV 01   740-045050   1E022240007    DC 52V Power Supply
Module
PSM 6                 REV 01   740-045050   1E02224001C    DC 52V Power Supply
Module
PSM 7                 REV 01   740-045050   1E02224001D    DC 52V Power Supply
Module
PSM 8                 REV 01   740-045050   1E02224001B    DC 52V Power Supply
Module
PDM 0                 REV 01   740-045234   1E262250067    DC Power Dist Module
Routing Engine 0      REV 02   740-041821   9009099704     RE-S-1800x4
Routing Engine 1      REV 02   740-041821   9009094146     RE-S-1800x4
CB 0                   REV 13   750-040257   CAAF8436       Control Board
CB 1                   REV 08   750-040257   CAAB3486       Control Board
SPMB 0                 REV 02   711-041855   ABBV3825       PMB Board
SPMB 1                 REV 02   711-041855   CAAA6147       PMB Board
SFB 0                  REV 05   711-044466   ABBX5687       Switch Fabric Board
SFB 1                  REV 05   711-044466   ABBX5609       Switch Fabric Board
SFB 2                  REV 05   711-044466   ABBX5675       Switch Fabric Board
SFB 3                  REV 05   711-044466   ABBX5701       Switch Fabric Board
SFB 4                  REV 03   711-044466   ABBV6789       Switch Fabric Board
SFB 5                  REV 05   711-044466   ABBX5608       Switch Fabric Board

```

SFB 6	REV 05	711-044466	ABBX5678	Switch Fabric Board
SFB 7	REV 05	711-044466	ABBX5666	Switch Fabric Board
FPC 0	REV 33	750-028467	CAAE5971	MPC 3D 16x 10GE
CPU	REV 11	711-029089	CAAF4257	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AD0915E000C	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AD0927A005R	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AD0915E000R	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD0927A005U	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AD0927A0061	SFP+-10G-SR
Xcvr 1	REV 01	740-021310	P09A02116	SFP+-10G-LRM
Xcvr 2	REV 01	740-021308	AD0927A0064	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD0927A006W	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AD0927A0065	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AD0927A0074	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AD0927A0062	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD0927A002K	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	AD0927A005G	SFP+-10G-SR
Xcvr 1	REV 01	740-021310	93D209A00002	SFP+-10G-LRM
Xcvr 2	REV 01	740-021308	AD0927A002N	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD0927A001W	SFP+-10G-SR
FPC 1	REV 18	750-033205	ZE0128	MPC Type 3
CPU	REV 06	711-035209	ZG5431	HMPC PMB 2G
MIC 0	REV 15	750-033199	ZP6435	1X100GE CFP
PIC 0		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0	REV 01	740-032210	J11E46118	CFP-100G-LR4
MIC 1	REV 15	750-033199	ZP6442	1X100GE CFP
PIC 2		BUILTIN	BUILTIN	1X100GE CFP
Xcvr 0	REV 01	740-032210	UMN03T4	CFP-100G-LR4
FPC 2	REV 21	750-033205	ZG5028	MPC Type 3
CPU	REV 05	711-035209	YX3911	HMPC PMB 2G
MIC 0	REV 03	750-036233	ZL2036	2X40GE QSFP
PIC 0		BUILTIN	BUILTIN	2X40GE QSFP
Xcvr 0	REV 01	740-032986	QB220708	QSFP+-40G-SR4
Xcvr 1	REV 01	740-032986	QB220735	QSFP+-40G-SR4
MIC 1	REV 03	750-036233	ZL2028	2X40GE QSFP
PIC 2		BUILTIN	BUILTIN	2X40GE QSFP
Xcvr 0	REV 01	740-032986	QB220727	QSFP+-40G-SR4
Xcvr 1	REV 01	740-032986	QB220715	QSFP+-40G-SR4
FPC 3	REV 33	750-028467	CAAF5400	MPC 3D 16x 10GE
CPU	REV 11	711-029089	CAAH7626	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	973152A00066	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	973152A00021	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	973152A00062	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	973152A00027	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	973152A00065	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	973152A00069	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	973152A00026	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	973152A00003	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	973152A00035	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	973152A00004	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	973152A00049	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	973152A00055	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+

Xcvr 0	REV 01	740-021308	973152A00010	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	973152A00001	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	973152A00073	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	973152A00012	SFP+-10G-SR
FPC 4	REV 24	750-033205	ZJ6564	MPC Type 3
CPU	REV 07	711-035209	ZK9031	HMPC PMB 2G
MIC 0	REV 03	750-036233	ZL2014	2X40GE QSFP
PIC 0		BUILTIN	BUILTIN	2X40GE QSFP
Xcvr 0	REV 01	740-032986	QB390978	QSFP+-40G-SR4
Xcvr 1	REV 01	740-032986	QB390950	QSFP+-40G-SR4
MIC 1	REV 03	750-036233	ZL2034	2X40GE QSFP
PIC 2		BUILTIN	BUILTIN	2X40GE QSFP
Xcvr 0	REV 01	740-032986	QB390970	QSFP+-40G-SR4
Xcvr 1	REV 01	740-032986	QB220712	QSFP+-40G-SR4
FPC 5	REV 05	750-037358	ZT0665	MPC Type 4-1
CPU	REV 07	711-035209	ZK3181	HMPC PMB 2G
FPC 6	REV 30	750-028467	ZM4986	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ZP6541	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
FPC 7	REV 33	750-028467	CAAF5552	MPC 3D 16x 10GE
CPU	REV 11	711-029089	CAAH7601	AMPC PMB
FPC 8	REV 03	750-045372	CAAD3111	MPC Type 3
CPU	REV 08	711-035209	CAAD8033	HMPC PMB 2G
MIC 0	REV 03	750-036233	ZL2032	2X40GE QSFP
PIC 0		BUILTIN	BUILTIN	2X40GE QSFP
MIC 1	REV 03	750-036233	ZL2021	2X40GE QSFP
PIC 2		BUILTIN	BUILTIN	2X40GE QSFP
FPC 9	REV 07	750-038491	ZV7947	MPCE Type 2 3D
CPU	REV 03	711-038484	ZV1360	MPCE PMB 2G
MIC 0	REV 02	750-028380	JR6609	3D 2x 10GE XFP
PIC 0		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0	REV 01	740-014289	T07M71204	XFP-10G-SR
PIC 1		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0	REV 01	740-014289	AD0932M004Y	XFP-10G-SR
MIC 1	REV 02	750-028380	JR6544	3D 2x 10GE XFP
PIC 2		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0	REV 01	740-014289	AD0932M0092	XFP-10G-SR
PIC 3		BUILTIN	BUILTIN	1x 10GE XFP
Xcvr 0	REV 01	740-014289	AD0932M000J	XFP-10G-SR
ADC 0	REV 13	750-043596	ABBX5532	Adapter Card
ADC 1	REV 13	750-043596	ABBX5550	Adapter Card
ADC 2	REV 13	750-043596	ABBX5556	Adapter Card
ADC 3	REV 13	750-043596	ABBX5568	Adapter Card
ADC 4	REV 13	750-043596	ABBX5571	Adapter Card
ADC 5	REV 13	750-043596	ABBX5553	Adapter Card
ADC 6	REV 13	750-043596	ABBX5578	Adapter Card
ADC 7	REV 13	750-043596	ABBX5541	Adapter Card
ADC 8	REV 13	750-043596	ABBX5560	Adapter Card
ADC 9	REV 07	750-043596	ABBV7188	Adapter Card
Fan Tray 0	REV 03	760-046960	ACAY0127	172mm FanTray - 6 Fans
Fan Tray 1	REV 2A	760-046960	ACAY0068	172mm FanTray - 6 Fans
Fan Tray 2	REV 2A	760-046960	ACAY0072	172mm FanTray - 6 Fans
Fan Tray 3	REV 2A	760-046960	ACAY0070	172mm FanTray - 6 Fans

To verify FPC status, use the **show chassis fpc** command. Sample command output from an MX2010 router is as follows:

```
user@host> show chassis fpc
```

Slot	Temp State	CPU Utilization (%)		Memory Interrupt	Utilization (%)		Buffer
		(C)	Total		DRAM (MB)	Heap	
0	Online	38	13	3	2048	18	13
1	Online	36	16	2	2048	14	14
2	Online	36	15	2	2048	14	14
3	Online	37	17	4	2048	18	13
4	Online	37	13	2	2048	14	14
5	Online	36	20	4	2048	17	14
6	Online	36	16	4	2048	18	13
7	Online	37	20	4	2048	18	13
8	Online	35	10	2	2048	14	14
9	Online	34	14	2	2048	11	13

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an MX2010 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
CPU temperature         35 degrees C / 95 degrees F
DRAM                   17152 MB
Memory utilization      12 percent
CPU utilization:
  User                  1 percent
  Background            0 percent
  Kernel                6 percent
  Interrupt             1 percent
  Idle                  92 percent
Model                  RE-S-1800x4
Serial ID               9009099704
Start time              2012-09-26 13:00:16 PDT
Uptime                  14 hours, 51 minutes, 38 seconds
Last reboot reason      Router rebooted after a normal shutdown.
Load averages:          1 minute  5 minute  15 minute
                        0.06       0.03     0.01

```

Routing Engine status:

Slot 1:

```

Current state           Backup
Election priority       Backup (default)
Temperature             33 degrees C / 91 degrees F
CPU temperature         33 degrees C / 91 degrees F
DRAM                   17152 MB
Memory utilization      11 percent
CPU utilization:
  User                  0 percent
  Background            0 percent
  Kernel                0 percent
  Interrupt             0 percent
  Idle                  100 percent
Model                  RE-S-1800x4
Serial ID               9009094146
Start time              2012-09-26 13:11:17 PDT
Uptime                  14 hours, 40 minutes, 24 seconds
Last reboot reason      Router rebooted after a normal shutdown.
Load averages:          1 minute  5 minute  15 minute
                        0.00       0.00     0.00

```

Table 89 on page 299 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of an MX2020 router.

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	1	1	0	Fan Tray 0 Fan 1	fan	0	Online	0	None	000000	33663469	14930619
4	1	2	0	Fan Tray 0 Fan 2	fan	0	Online	0	None	000000	33663469	14930619
4	1	3	0	Fan Tray 0 Fan 3	fan	0	Online	0	None	000000	33663469	14930619
4	1	4	0	Fan Tray 0 Fan 4	fan	0	Online	0	None	000000	33663469	14930619
4	1	5	0	Fan Tray 0 Fan 5	fan	0	Online	0	None	000000	33663469	14930619
4	1	6	0	Fan Tray 0 Fan 6	fan	0	Online	0	None	000000	33663469	14930620
4	2	1	0	Fan Tray 1 Fan 1	fan	1	Online	0	None	000000	33663473	14930616
4	2	2	0	Fan Tray 1 Fan 2	fan	1	Online	0	None	000000	33663473	14930616
4	2	3	0	Fan Tray 1 Fan 3	fan	1	Online	0	None	000000	33663473	14930616
4	2	4	0	Fan Tray 1 Fan 4	fan	1	Online	0	None	000000	33663473	14930616
4	2	5	0	Fan Tray 1 Fan 5	fan	1	Online	0	None	000000	33663473	14930616
4	2	6	0	Fan Tray 1 Fan 6	fan	1	Online	0	None	000000	33663473	14930616
4	3	1	0	Fan Tray 2 Fan 1	fan	2	Online	0	None	000000	33663477	14930612
4	3	2	0	Fan Tray 2 Fan 2	fan	2	Online	0	None	000000	33663477	14930613
4	3	3	0	Fan Tray 2 Fan 3	fan	2	Online	0	None	000000	33663477	14930613
4	3	4	0	Fan Tray 2 Fan 4	fan	2	Online	0	None	000000	33663477	14930613

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	3	5	0	Fan Tray 2 Fan 5	fan	2	Online	0	None	0000000	33663477	1493063
4	3	6	0	Fan Tray 2 Fan 6	fan	2	Online	0	None	0000000	33663477	1493063
4	4	1	0	Fan Tray 3 Fan 1	fan	3	Online	0	None	0000000	33663482	1493069
4	4	2	0	Fan Tray 3 Fan 2	fan	3	Online	0	None	0000000	33663482	1493069
4	4	3	0	Fan Tray 3 Fan 3	fan	3	Online	0	None	0000000	33663482	1493060
4	4	4	0	Fan Tray 3 Fan 4	fan	3	Online	0	None	0000000	33663482	1493060
4	4	5	0	Fan Tray 3 Fan 5	fan	3	Online	0	None	0000000	33663482	1493060
4	4	6	0	Fan Tray 3 Fan 6	fan	3	Online	0	None	0000000	33663482	1493060
7	1	0	0	FPC: MPC 3D16x10GE @ 0/*/*	flexiblePicConcentrator	0	Online	65	None	0000000	3368054	14913938
7	2	0	0	FPC: MPC 3D16x10GE @ 1/*/*	flexiblePicConcentrator	1	Online	64	None	0000000	33681028	14913065
7	3	0	0	FPC: MPC 3D16x10GE @ 2/*/*	flexiblePicConcentrator	2	Online	63	None	0000000	33681945	14912147
7	4	0	0	FPC: MPC 3D16x10GE @ 3/*/*	flexiblePicConcentrator	3	Online	64	None	0000000	3368269	14911464
7	5	0	0	FPC: MPC 3D16x10GE @ 4/*/*	flexiblePicConcentrator	4	Online	64	None	0000000	33683524	14910569
7	6	0	0	FPC: MPC 3D16x10GE @ 5/*/*	flexiblePicConcentrator	5	Online	64	None	0000000	33684387	14909706
7	7	0	0	FPC: MPC 3D16x10GE @ 6/*/*	flexiblePicConcentrator	6	Online	64	None	0000000	33685048	14909045

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	8	0	0	FPC: MPC 3D16x10GE @ 7/*/*	flexiblePicConcentrator	7	Online	64	None	0000000	33685948	14908145
7	9	0	0	FPC: MPC 3D16x10GE @ 8/*/*	flexiblePicConcentrator	8	Online	64	None	0000000	33686881	14907264
7	10	0	0	FPC: MPC 3D16x10GE @ 9/*/*	flexiblePicConcentrator	9	Online	64	None	0000000	33687754	14906340
7	11	0	0	FPC: MPC 3D16x10GE @ 10/*/*	flexiblePicConcentrator	10	Online	66	None	0000000	33688444	14905681
7	12	0	0	FPC: MPC 3D16x10GE @ 11/*/*	flexiblePicConcentrator	11	Online	63	None	0000000	33689276	14904819
7	13	0	0	FPC: MPC 3D16x10GE @ 12/*/*	flexiblePicConcentrator	12	Online	63	None	0000000	33690152	14903942
7	14	0	0	FPC: MPC 3D16x10GE @ 13/*/*	flexiblePicConcentrator	13	Online	62	None	0000000	33691035	14903060
7	15	0	0	FPC: MPC 3D16x10GE @ 14/*/*	flexiblePicConcentrator	14	Online	64	None	0000000	33691955	14902140
7	16	0	0	FPC: MPC 3D16x10GE @ 15/*/*	flexiblePicConcentrator	15	Online	63	None	0000000	33692833	14901262
7	17	0	0	FPC: MPC 3D16x10GE @ 16/*/*	flexiblePicConcentrator	16	Online	64	None	0000000	33693530	14900348
7	18	0	0	FPC: MPC 3D16x10GE @ 17/*/*	flexiblePicConcentrator	17	Online	63	None	0000000	33694638	14899464
7	19	0	0	FPC: MPC 3D16x10GE @ 18/*/*	flexiblePicConcentrator	18	Online	64	None	0000000	33695525	14898572
7	20	0	0	FPC: MPC 3D16x10GE @ 19/*/*	flexiblePicConcentrator	19	Online	65	None	0000000	33696386	14897711

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	1	1	0	PIC: 4x 10GE(LAN) SFP+ @ 0/0/*	portInterfaceCard	0	Online	0	None	0000000	33700132	14899965
8	1	2	0	PIC: 4x 10GE(LAN) SFP+ @ 0/1/*	portInterfaceCard	0	Online	0	None	0000000	33700132	14899966
8	1	3	0	PIC: 4x 10GE(LAN) SFP+ @ 0/2/*	portInterfaceCard	0	Online	0	None	0000000	33700132	14899966
8	1	4	0	PIC: 4x 10GE(LAN) SFP+ @ 0/3/*	portInterfaceCard	0	Online	0	None	0000000	33700133	14899965
8	2	1	0	PIC: 4x 10GE(LAN) SFP+ @ 1/0/*	portInterfaceCard	1	Online	0	None	0000000	33702764	14891334
8	2	2	0	PIC: 4x 10GE(LAN) SFP+ @ 1/1/*	portInterfaceCard	1	Online	0	None	0000000	33702764	14891334
8	2	3	0	PIC: 4x 10GE(LAN) SFP+ @ 1/2/*	portInterfaceCard	1	Online	0	None	0000000	33702764	14891334
8	2	4	0	PIC: 4x 10GE(LAN) SFP+ @ 1/3/*	portInterfaceCard	1	Online	0	None	0000000	33702764	14891334
8	3	1	0	PIC: 4x 10GE(LAN) SFP+ @ 2/0/*	portInterfaceCard	2	Online	0	None	0000000	33701875	14892223
8	3	2	0	PIC: 4x 10GE(LAN) SFP+ @ 2/1/*	portInterfaceCard	2	Online	0	None	0000000	33701875	14892223

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	3	0	PIC: 4x 10GE(LAN) SFP+ @ 2/2/*	portInterfaceCard	2	Online	0	None	00000000	33701875	14892223
8	3	4	0	PIC: 4x 10GE(LAN) SFP+ @ 2/3/*	portInterfaceCard	2	Online	0	None	00000000	33701875	14892223
8	4	1	0	PIC: 4x 10GE(LAN) SFP+ @ 3/0/*	portInterfaceCard	3	Online	0	None	00000000	33702903	14891197
8	4	2	0	PIC: 4x 10GE(LAN) SFP+ @ 3/1/*	portInterfaceCard	3	Online	0	None	00000000	33702903	14891197
8	4	3	0	PIC: 4x 10GE(LAN) SFP+ @ 3/2/*	portInterfaceCard	3	Online	0	None	00000000	33702903	14891197
8	4	4	0	PIC: 4x 10GE(LAN) SFP+ @ 3/3/*	portInterfaceCard	3	Online	0	None	00000000	33702903	14891198
8	5	1	0	PIC: 4x 10GE(LAN) SFP+ @ 4/0/*	portInterfaceCard	4	Online	0	None	00000000	33703744	14890856
8	5	2	0	PIC: 4x 10GE(LAN) SFP+ @ 4/1/*	portInterfaceCard	4	Online	0	None	00000000	33703744	14890856
8	5	3	0	PIC: 4x 10GE(LAN) SFP+ @ 4/2/*	portInterfaceCard	4	Online	0	None	00000000	33703744	14890856
8	5	4	0	PIC: 4x 10GE(LAN) SFP+ @ 4/3/*	portInterfaceCard	4	Online	0	None	00000000	33703744	14890357

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	6	1	0	PIC: 4x 10GE(LAN) SFP+ @ 5/0/*	portInterfaceCard	5	Online	0	None	0000000	33704471	14889630
8	6	2	0	PIC: 4x 10GE(LAN) SFP+ @ 5/1/*	portInterfaceCard	5	Online	0	None	0000000	33704471	14889630
8	6	3	0	PIC: 4x 10GE(LAN) SFP+ @ 5/2/*	portInterfaceCard	5	Online	0	None	0000000	33704471	14889630
8	6	4	0	PIC: 4x 10GE(LAN) SFP+ @ 5/3/*	portInterfaceCard	5	Online	0	None	0000000	33704471	14889630
8	7	1	0	PIC: 4x 10GE(LAN) SFP+ @ 6/0/*	portInterfaceCard	6	Online	0	None	0000000	33707121	14886982
8	7	2	0	PIC: 4x 10GE(LAN) SFP+ @ 6/1/*	portInterfaceCard	6	Online	0	None	0000000	33707121	14886982
8	7	3	0	PIC: 4x 10GE(LAN) SFP+ @ 6/2/*	portInterfaceCard	6	Online	0	None	0000000	33707121	14886982
8	7	4	0	PIC: 4x 10GE(LAN) SFP+ @ 6/3/*	portInterfaceCard	6	Online	0	None	0000000	33707121	14886982
8	8	1	0	PIC: 4x 10GE(LAN) SFP+ @ 7/0/*	portInterfaceCard	7	Online	0	None	0000000	33707743	14886360
8	8	2	0	PIC: 4x 10GE(LAN) SFP+ @ 7/1/*	portInterfaceCard	7	Online	0	None	0000000	33707779	14886324

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	3	0	PIC: 4x 10GE(LAN) SFP+ @ 7/2/*	portInterfaceCard	7	Online	0	None	0000000	3370779	1488633
8	8	4	0	PIC: 4x 10GE(LAN) SFP+ @ 7/3/*	portInterfaceCard	7	Online	0	None	0000000	3370780	1488634
8	9	1	0	PIC: 4x 10GE(LAN) SFP+ @ 8/0/*	portInterfaceCard	8	Online	0	None	0000000	33708629	1488545
8	9	2	0	PIC: 4x 10GE(LAN) SFP+ @ 8/1/*	portInterfaceCard	8	Online	0	None	0000000	33708659	14885444
8	9	3	0	PIC: 4x 10GE(LAN) SFP+ @ 8/2/*	portInterfaceCard	8	Online	0	None	0000000	33708659	14885444
8	9	4	0	PIC: 4x 10GE(LAN) SFP+ @ 8/3/*	portInterfaceCard	8	Online	0	None	0000000	33708660	14885444
8	10	1	0	PIC: 4x 10GE(LAN) SFP+ @ 9/0/*	portInterfaceCard	9	Online	0	None	0000000	33709080	14885024
8	10	2	0	PIC: 4x 10GE(LAN) SFP+ @ 9/1/*	portInterfaceCard	9	Online	0	None	0000000	33709080	14885024
8	10	3	0	PIC: 4x 10GE(LAN) SFP+ @ 9/2/*	portInterfaceCard	9	Online	0	None	0000000	33709081	14885024
8	10	4	0	PIC: 4x 10GE(LAN) SFP+ @ 9/3/*	portInterfaceCard	9	Online	0	None	0000000	33709081	14885024

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	11	1	0	PIC: 4x 10GE(LAN) SFP+ @ 10/0/*	portInterfaceCard	10	Online	0	None	00000000	33710044	14884061
8	11	2	0	PIC: 4x 10GE(LAN) SFP+ @ 10/1/*	portInterfaceCard	10	Online	0	None	00000000	33710045	14884060
8	11	3	0	PIC: 4x 10GE(LAN) SFP+ @ 10/2/*	portInterfaceCard	10	Online	0	None	00000000	33710045	14884061
8	11	4	0	PIC: 4x 10GE(LAN) SFP+ @ 10/3/*	portInterfaceCard	10	Online	0	None	00000000	33710045	14884061
8	12	1	0	PIC: 4x 10GE(LAN) SFP+ @ 11/0/*	portInterfaceCard	11	Online	0	None	00000000	33709791	14884315
8	12	2	0	PIC: 4x 10GE(LAN) SFP+ @ 11/1/*	portInterfaceCard	11	Online	0	None	00000000	33709791	14884315
8	12	3	0	PIC: 4x 10GE(LAN) SFP+ @ 11/2/*	portInterfaceCard	11	Online	0	None	00000000	33709791	14884315
8	12	4	0	PIC: 4x 10GE(LAN) SFP+ @ 11/3/*	portInterfaceCard	11	Online	0	None	00000000	33709792	14884315
8	13	1	0	PIC: 4x 10GE(LAN) SFP+ @ 12/0/*	portInterfaceCard	12	Online	0	None	00000000	33712073	14882034
8	13	2	0	PIC: 4x 10GE(LAN) SFP+ @ 12/1/*	portInterfaceCard	12	Online	0	None	00000000	33712073	14882033

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	13	3	0	PIC: 4x 10GE(LAN) SFP+ @ 12/2/*	portInterfaceCard	12	Online	0	None	0000000	33712074	14882034
8	13	4	0	PIC: 4x 10GE(LAN) SFP+ @ 12/3/*	portInterfaceCard	12	Online	0	None	0000000	33712074	14882034
8	14	1	0	PIC: 4x 10GE(LAN) SFP+ @ 13/0/*	portInterfaceCard	13	Online	0	None	0000000	33711267	14882840
8	14	2	0	PIC: 4x 10GE(LAN) SFP+ @ 13/1/*	portInterfaceCard	13	Online	0	None	0000000	33711267	14882840
8	14	3	0	PIC: 4x 10GE(LAN) SFP+ @ 13/2/*	portInterfaceCard	13	Online	0	None	0000000	33711267	14882840
8	14	4	0	PIC: 4x 10GE(LAN) SFP+ @ 13/3/*	portInterfaceCard	13	Online	0	None	0000000	33711268	14882841
8	15	1	0	PIC: 4x 10GE(LAN) SFP+ @ 14/0/*	portInterfaceCard	14	Online	0	None	0000000	33714106	14880008
8	15	2	0	PIC: 4x 10GE(LAN) SFP+ @ 14/1/*	portInterfaceCard	14	Online	0	None	0000000	33714153	14879956
8	15	3	0	PIC: 4x 10GE(LAN) SFP+ @ 14/2/*	portInterfaceCard	14	Online	0	None	0000000	33714194	14879915
8	15	4	0	PIC: 4x 10GE(LAN) SFP+ @ 14/3/*	portInterfaceCard	14	Online	0	None	0000000	33714195	14879914

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	16	1	0	PIC: 4x 10GE(LAN) SFP+ @ 15/0/*	portInterfaceCard	15	Online	0	None	00000000	3375046	14879063
8	16	2	0	PIC: 4x 10GE(LAN) SFP+ @ 15/1/*	portInterfaceCard	15	Online	0	None	00000000	3375046	14879063
8	16	3	0	PIC: 4x 10GE(LAN) SFP+ @ 15/2/*	portInterfaceCard	15	Online	0	None	00000000	3375046	14879063
8	16	4	0	PIC: 4x 10GE(LAN) SFP+ @ 15/3/*	portInterfaceCard	15	Online	0	None	00000000	3375046	14879063
8	17	1	0	PIC: 4x 10GE(LAN) SFP+ @ 16/0/*	portInterfaceCard	16	Online	0	None	00000000	33713987	14880124
8	17	2	0	PIC: 4x 10GE(LAN) SFP+ @ 16/1/*	portInterfaceCard	16	Online	0	None	00000000	33713987	14880124
8	17	3	0	PIC: 4x 10GE(LAN) SFP+ @ 16/2/*	portInterfaceCard	16	Online	0	None	00000000	33713987	14880124
8	17	4	0	PIC: 4x 10GE(LAN) SFP+ @ 16/3/*	portInterfaceCard	16	Online	0	None	00000000	33713987	14880124
8	18	1	0	PIC: 4x 10GE(LAN) SFP+ @ 17/0/*	portInterfaceCard	17	Online	0	None	00000000	33716606	1487505
8	18	2	0	PIC: 4x 10GE(LAN) SFP+ @ 17/1/*	portInterfaceCard	17	Online	0	None	00000000	33716606	1487505

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	18	3	0	PIC: 4x 10GE(LAN) SFP+ @ 17/2/*	portInterfaceCard	17	Online	0	None	00000000	33716606	1487505
8	18	4	0	PIC: 4x 10GE(LAN) SFP+ @ 17/3/*	portInterfaceCard	17	Online	0	None	00000000	33716606	1487505
8	19	1	0	PIC: 4x 10GE(LAN) SFP+ @ 18/0/*	portInterfaceCard	18	Online	0	None	00000000	33715933	14878179
8	19	2	0	PIC: 4x 10GE(LAN) SFP+ @ 18/1/*	portInterfaceCard	18	Online	0	None	00000000	33715933	14878179
8	19	3	0	PIC: 4x 10GE(LAN) SFP+ @ 18/2/*	portInterfaceCard	18	Online	0	None	00000000	33715933	14878179
8	19	4	0	PIC: 4x 10GE(LAN) SFP+ @ 18/3/*	portInterfaceCard	18	Online	0	None	00000000	33715933	14878179
8	20	1	0	PIC: 4x 10GE(LAN) SFP+ @ 19/0/*	portInterfaceCard	19	Online	0	None	00000000	33718181	14875932
8	20	2	0	PIC: 4x 10GE(LAN) SFP+ @ 19/1/*	portInterfaceCard	19	Online	0	None	00000000	33718181	14875932
8	20	3	0	PIC: 4x 10GE(LAN) SFP+ @ 19/2/*	portInterfaceCard	19	Online	0	None	00000000	33718181	14875932
8	20	4	0	PIC: 4x 10GE(LAN) SFP+ @ 19/3/*	portInterfaceCard	19	Online	0	None	00000000	33718181	14875932
9	1	0	0	Routing Engine 0	routingEngine	0	Online	62	None	00000000	33663298	14930816

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
9	2	0	0	Routing Engine 1	routingEngine	1	Online	62	None	0000000	33668298	14930816
10	1	1	0	FPM Board	frontPanelModule	0	Online	0	None	0000000	33668356	14930888
12	1	0	0	CB 0	controlBoard	0	Online	66	None	0000000	33668573	14930541
12	2	0	0	CB 1	controlBoard	1	Online	65	None	0000000	33668588	14930526
14	1	0	0	SPMB 0	processorMezzanineBoardForSIB	0	Online	60	None	0000000	33668589	14927726
14	2	0	0	SPMB 1	processorMezzanineBoardForSIB	1	Online	60	None	0000000	33668602	14927913
15	1	0	0	SFB 0	switchFabricBoard	0	Online	67	None	0000000	33668737	14925378
15	2	0	0	SFB 1	switchFabricBoard	1	Online	66	None	0000000	33669807	14924308
15	3	0	0	SFB 2	switchFabricBoard	2	Online	67	None	0000000	33670878	14923238
15	4	0	0	SFB 3	switchFabricBoard	3	Online	66	None	0000000	33671949	14922167
15	5	0	0	SFB 4	switchFabricBoard	4	Online	67	None	0000000	33673020	14921097
15	6	0	0	SFB 5	switchFabricBoard	5	Online	68	None	0000000	33674092	14920025
15	7	0	0	SFB 6	switchFabricBoard	6	Online	68	None	0000000	33675163	14918953
15	8	0	0	SFB 7	switchFabricBoard	7	Online	67	None	0000000	33676236	14917881
20	1	1	0	MIC: @ 0/0/*	portInterfaceCard	0	Empty	0	None	0000000	0000000	0
20	1	2	0	MIC: @ 0/1/*	portInterfaceCard	0	Empty	0	None	0000000	0000000	0
20	2	1	0	MIC: @ 1/0/*	portInterfaceCard	1	Empty	0	None	0000000	0000000	0
20	2	2	0	MIC: @ 1/1/*	portInterfaceCard	1	Empty	0	None	0000000	0000000	0
20	3	1	0	MIC: @ 2/0/*	portInterfaceCard	2	Empty	0	None	0000000	0000000	0
20	3	2	0	MIC: @ 2/1/*	portInterfaceCard	2	Empty	0	None	0000000	0000000	0

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
20	4	1	0	MIC: @ 3/0/*	portInterfaceCard	3	Empty	0	None			0
20	4	2	0	MIC: @ 3/1/*	portInterfaceCard	3	Empty	0	None			0
20	5	1	0	MIC: @ 4/0/*	portInterfaceCard	4	Empty	0	None			0
20	5	2	0	MIC: @ 4/1/*	portInterfaceCard	4	Empty	0	None			0
20	6	1	0	MIC: @ 5/0/*	portInterfaceCard	5	Empty	0	None			0
20	6	2	0	MIC: @ 5/1/*	portInterfaceCard	5	Empty	0	None			0
20	7	1	0	MIC: @ 6/0/*	portInterfaceCard	6	Empty	0	None			0
20	7	2	0	MIC: @ 6/1/*	portInterfaceCard	6	Empty	0	None			0
20	8	1	0	MIC: @ 7/0/*	portInterfaceCard	7	Empty	0	None			0
20	8	2	0	MIC: @ 7/1/*	portInterfaceCard	7	Empty	0	None			0
20	9	1	0	MIC: @ 8/0/*	portInterfaceCard	8	Empty	0	None			0
20	9	2	0	MIC: @ 8/1/*	portInterfaceCard	8	Empty	0	None			0
20	10	1	0	MIC: @ 9/0/*	portInterfaceCard	9	Empty	0	None			0
20	10	2	0	MIC: @ 9/1/*	portInterfaceCard	9	Empty	0	None			0
20	11	1	0	MIC: @ 10/0/*	portInterfaceCard	10	Empty	0	None			0
20	11	2	0	MIC: @ 10/1/*	portInterfaceCard	10	Empty	0	None			0
20	12	1	0	MIC: @ 11/0/*	portInterfaceCard	11	Empty	0	None			0

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
20	12	2	0	MIC: @ 11/1/*	portInterfaceCard	11	Empty	0	None	000000	000000	0
20	13	1	0	MIC: @ 12/0/*	portInterfaceCard	12	Empty	0	None	000000	000000	0
20	13	2	0	MIC: @ 12/1/*	portInterfaceCard	12	Empty	0	None	000000	000000	0
20	14	1	0	MIC: @ 13/0/*	portInterfaceCard	13	Empty	0	None	000000	000000	0
20	14	2	0	MIC: @ 13/1/*	portInterfaceCard	13	Empty	0	None	000000	000000	0
20	15	1	0	MIC: @ 14/0/*	portInterfaceCard	14	Empty	0	None	000000	000000	0
20	15	2	0	MIC: @ 14/1/*	portInterfaceCard	14	Empty	0	None	000000	000000	0
20	16	1	0	MIC: @ 15/0/*	portInterfaceCard	15	Empty	0	None	000000	000000	0
20	16	2	0	MIC: @ 15/1/*	portInterfaceCard	15	Empty	0	None	000000	000000	0
20	17	1	0	MIC: @ 16/0/*	portInterfaceCard	16	Empty	0	None	000000	000000	0
20	17	2	0	MIC: @ 16/1/*	portInterfaceCard	16	Empty	0	None	000000	000000	0
20	18	1	0	MIC: @ 17/0/*	portInterfaceCard	17	Empty	0	None	000000	000000	0
20	18	2	0	MIC: @ 17/1/*	portInterfaceCard	17	Empty	0	None	000000	000000	0
20	19	1	0	MIC: @ 18/0/*	portInterfaceCard	18	Empty	0	None	000000	000000	0
20	19	2	0	MIC: @ 18/1/*	portInterfaceCard	18	Empty	0	None	000000	000000	0
20	20	1	0	MIC: @ 19/0/*	portInterfaceCard	19	Empty	0	None	000000	000000	0
20	20	2	0	MIC: @ 19/1/*	portInterfaceCard	19	Empty	0	None	000000	000000	0

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
21	1	0	0	PDM 0	powerDistributionModule	0	Online	0	None	000000	33663469	14930657
21	2	0	0	PDM 1	powerDistributionModule	1	Online	0	None	000000	33663469	14930658
21	3	0	0	PDM 2	powerDistributionModule	2	Online	0	None	000000	33663469	14930658
21	4	0	0	PDM 3	powerDistributionModule	3	Online	0	None	000000	33663469	14930658
22	1	0	0	PSM 0	powerSupplyModule	0	Online	67	None	000000	33663466	14930662
22	2	0	0	PSM 1	powerSupplyModule	1	Online	65	None	000000	33663466	14930662
22	3	0	0	PSM 2	powerSupplyModule	2	Online	64	None	000000	33663466	14930662
22	4	0	0	PSM 3	powerSupplyModule	3	Online	64	None	000000	33663466	14930662
22	5	0	0	PSM 4	powerSupplyModule	4	Online	65	None	000000	33663467	14930662
22	6	0	0	PSM 5	powerSupplyModule	5	Online	0	None	000000	000000	0
22	7	0	02	PSM 6	powerSupplyModule	6	Online	0	None	000000	000000	0
22	8	0	0	PSM 7	powerSupplyModule	7	Online	70	None	000000	33663467	14930662
22	9	0	0	PSM 8	powerSupplyModule	8	Online	65	None	000000	33663467	14930663
22	10	0	0	PSM 9	powerSupplyModule	9	Online	65	None	000000	33663467	14930663
22	11	0	0	PSM 10	powerSupplyModule	10	Online	63	None	000000	33663467	14930663
22	12	0	0	PSM 11	powerSupplyModule	11	Online	63	None	000000	33663468	14930663
22	13	0	0	PSM 12	powerSupplyModule	12	Online	64	None	000000	33663468	14930663
22	14	0	0	PSM 13	powerSupplyModule	13	Online	64	None	000000	33663468	14930663
22	15	0	0	PSM 14	powerSupplyModule	14	Online	64	None	000000	33663468	14930663
22	16	0	0	PSM 15	powerSupplyModule	15	Online	65	None	000000	33663468	14930663
22	17	0	0	PSM 16	powerSupplyModule	16	Online	64	None	000000	33663469	14930663
22	18	0	0	PSM 17	powerSupplyModule	17	Online	64	None	000000	33663469	14930663
23	1	0	0	ADC 0	adapterCard	0	Online	71	None	000000	33680088	14914050
23	2	0	0	ADC 1	adapterCard	1	Online	71	None	000000	33680086	14913247

Table 89: jnxFruContents Objects in the jnxFruTable of an MX2020 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
23	3	0	0	ADC 2	adapterCard	2	Online	70	None	00000000	33681713	14912420
23	4	0	0	ADC 3	adapterCard	3	Online	71	None	00000000	33682524	14911610
23	5	0	0	ADC 4	adapterCard	4	Online	71	None	00000000	33683345	14910789
23	6	0	0	ADC 5	adapterCard	5	Online	71	None	00000000	33684176	14909958
23	7	0	0	ADC 6	adapterCard	6	Online	70	None	00000000	33685009	14909126
23	8	0	0	ADC 7	adapterCard	7	Online	72	None	00000000	33685837	14908298
23	9	0	0	ADC 8	adapterCard	8	Online	71	None	00000000	33686659	14907456
23	10	0	0	ADC 9	adapterCard	9	Online	72	None	00000000	33687521	14906614
23	11	0	0	ADC 10	adapterCard	10	Online	71	None	00000000	33688370	14905766
23	12	0	0	ADC 11	adapterCard	11	Online	73	None	00000000	33689218	14904918
23	13	0	0	ADC 12	adapterCard	12	Online	72	None	00000000	33690086	14904050
23	14	0	0	ADC 13	adapterCard	13	Online	75	None	00000000	33690988	14903202
23	15	0	0	ADC 14	adapterCard	14	Online	74	None	00000000	33691795	14902341
23	16	0	0	ADC 15	adapterCard	15	Online	75	None	00000000	33692667	14901469
23	17	0	0	ADC 16	adapterCard	16	Online	74	None	00000000	33693538	14900608
23	18	0	0	ADC 17	adapterCard	17	Online	73	None	00000000	33694416	14899721
23	19	0	0	ADC 18	adapterCard	18	Online	73	None	00000000	33695300	14898888
23	20	0	0	ADC 19	adapterCard	19	Online	72	None	00000000	33696184	14897954

To verify L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample output from an MX2020 router is as follows:

```
user@host> show chassis hardware
```

```
Hardware inventory:
```

Item	Version	Part number	Serial number	Description
Chassis			JN11E2227AFJ	MX2020
Midplane	REV 27	750-040240	ABAB9384	Lower Power Midplane
Midplane 1	REV 04	711-032386	ABAB9386	Upper Backplane
PMP 1	REV 05	711-032428	ACAJ1579	Upper Power Midplane
PMP 0	REV 04	711-032426	ACAJ1524	Lower Power Midplane
FPM Board	REV 06	760-040242	ABBT8837	Front Panel Display

PSM 0 Module	REV 01	740-045050	1E022240056	DC 52V Power Supply
PSM 1 Module	REV 01	740-045050	1E022240054	DC 52V Power Supply
PSM 2 Module	REV 01	740-045050	1E02224005H	DC 52V Power Supply
PSM 3 Module	REV 01	740-045050	1E022240053	DC 52V Power Supply
PSM 4 Module	REV 01	740-045050	1E02224004K	DC 52V Power Supply
PSM 7 Module	REV 01	740-045050	1E02224006W	DC 52V Power Supply
PSM 8 Module	REV 01	740-045050	1E022240062	DC 52V Power Supply
PSM 9 Module	REV 01	740-045050	1E02224005B	DC 52V Power Supply
PSM 10 Module	REV 01	740-045050	1E02224005A	DC 52V Power Supply
PSM 11 Module	REV 01	740-045050	1E022240052	DC 52V Power Supply
PSM 12 Module	REV 01	740-045050	1E022240051	DC 52V Power Supply
PSM 13 Module	REV 01	740-045050	1E022240058	DC 52V Power Supply
PSM 14 Module	REV 01	740-045050	1E02224004L	DC 52V Power Supply
PSM 15 Module	REV 01	740-045050	1E02224005M	DC 52V Power Supply
PSM 16 Module	REV 01	740-045050	1E02224006S	DC 52V Power Supply
PSM 17 Module	REV 01	740-045050	1E02224005Z	DC 52V Power Supply
PDM 0	REV 01	740-045234	1E012150033	DC Power Dist Module
PDM 1	REV 01	740-045234	1E012150027	DC Power Dist Module
PDM 2	REV 01	740-045234	1E012150028	DC Power Dist Module
PDM 3	REV 01	740-045234	1E012150045	DC Power Dist Module
Routing Engine 0	REV 02	740-041821	9009089704	RE-S-1800x4
Routing Engine 1	REV 02	740-041821	9009094138	RE-S-1800x4
CB 0	REV 14	750-040257	CAAF8430	Control Board
CB 1	REV 08	750-040257	CAAB3482	Control Board
SPMB 0	REV 01	711-041855	ZS2290	PMB Board
SPMB 1	REV 02	711-041855	CAAA6141	PMB Board
SFB 0	REV 03	711-044466	ABBV6790	Switch Fabric Board
SFB 1	REV 03	711-044466	ABBV6806	Switch Fabric Board
SFB 2	REV 03	711-044466	ABBV6800	Switch Fabric Board
SFB 3	REV 03	711-044466	ABBV6799	Switch Fabric Board
SFB 4	REV 03	711-044466	ABBV6791	Switch Fabric Board
SFB 5	REV 03	711-044466	ABBV6817	Switch Fabric Board
SFB 6	REV 03	711-044466	ABBV6805	Switch Fabric Board
SFB 7	REV 03	711-044466	ABBV6781	Switch Fabric Board
FPC 0	REV 30	750-028467	ABBN0284	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN0507	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E00990	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E04357	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01327	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E04375	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02760	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02904	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E03963	SFP+-10G-USR

Xcvr 3	REV 01	740-030658	B11E00756	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04418	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01077	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01128	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01253	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E01140	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11F01626	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01075	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E01177	SFP+-10G-USR
FPC 1	REV 30	750-028467	ABBN0208	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABB11084	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04745	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11F01570	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E04388	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01439	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04739	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11F01869	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01675	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01901	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01346	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11F01288	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01824	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E04312	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02811	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E03847	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01495	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01265	SFP+-10G-USR
FPC 2	REV 30	750-028467	ZM5111	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ZP6607	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LJA	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MFZ	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NKL	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80KF4	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80FBJ	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MM2	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LJV	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NXV	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80N1H	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NLS	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80FL5	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NL9	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NG2	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80KDU	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80MG1	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80MM0	SFP+-10G-SR
FPC 3	REV 30	750-028467	ABBN0302	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN0495	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01581	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01176	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01251	SFP+-10G-USR

Xcvr 3	REV 01	740-030658	B11E02752	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E00786	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01020	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01023	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E02819	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02812	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11D04437	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01279	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01333	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E00978	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01018	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01784	SFP+-10G-USR
Xcvr 3	REV 01	740-031980	AK8ONKP	SFP+-10G-SR
FPC 4	REV 30	750-028467	ABBN0308	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABB11095	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04305	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01147	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01195	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01743	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01892	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02880	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E00725	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E01057	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02816	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11C04501	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E02764	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E00789	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01250	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02847	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E00787	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E03803	SFP+-10G-USR
FPC 5	REV 30	750-028467	ABBN0316	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABB11082	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00523	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K01848	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01865	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K00540	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00422	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K00428	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K00423	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K01855	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K01847	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K00526	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K00529	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K00525	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00425	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K00530	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01851	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K00528	SFP+-10G-SR
FPC 6	REV 32	750-028467	ABBN6832	MPC 3D 16x 10GE

CPU	REV 10	711-029089	ABBK6534	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80MB4	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80FQ6	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80N1F	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NLQ	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80KDR	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80FGJ	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80N5G	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80KD8	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LET	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80N1X	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NRF	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NL2	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80N3D	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MRB	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LEQ	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LER	SFP+-10G-SR
FPC 7	REV 32	750-028467	ABBN6811	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN7288	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NK8	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80LJG	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LBU	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80N21	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LEU	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NLM	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NL6	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LES	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LEN	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80ME0	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LMG	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80MM1	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80MG7	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80KF9	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NRQ	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NLE	SFP+-10G-SR
FPC 8	REV 23	750-028467	YN2982	MPC 3D 16x 10GE
CPU	REV 10	711-029089	YF8633	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	183363A00851	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A00772	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	183363A00875	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	183363A00882	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	183363A00735	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A00726	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	183363A00169	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	183363A00077	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	183363A00168	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A00676	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	183363A00732	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	183363A00091	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+

Xcvr 0	REV 01	740-031980	183363A00642	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A00725	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	183363A00871	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	183363A00853	SFP+-10G-SR
FPC 9	REV 32	750-028467	ABBN6798	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBK6556	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	9ZDZ06A00055	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	183363A00239	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	AD0915E003K	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD0915E003A	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80MRC	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NL5	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NKN	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80N3U	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80N1T	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AJ808DJ	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NG4	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80FND	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80FKQ	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NLT	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NKR	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LKM	SFP+-10G-SR
FPC 10	REV 32	750-028467	ABBN6813	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBK6542	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NA3	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NLF	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80MRH	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80KE4	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-021308	973152A00030	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80L9H	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80ME8	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NLR	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NG1	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MCA	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LFC	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LEM	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80N9X	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80LAC	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80LF2	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80N8T	SFP+-10G-SR
FPC 11	REV 30	750-028467	ABBN0281	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN0526	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01326	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E03973	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E00950	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E00674	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E00775	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E04461	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01074	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E02821	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+

Xcvr 0	REV 01	740-030658	B11E04501	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E00757	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01623	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E01022	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04359	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02751	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E02736	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E01178	SFP+-10G-USR
FPC 12	REV 32	750-028467	ABBN6796	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN7259	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K01856	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K01853	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01863	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02863	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02668	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02881	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A01671	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02627	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02725	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02692	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02730	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03081	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02736	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02568	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02747	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02579	SFP+-10G-SR
FPC 13	REV 30	750-028467	ABBN0270	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBJ0966	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NL1	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NXW	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80KD2	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80FMD	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80NKQ	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MGH	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80N38	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80NL7	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LEL	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80NKD	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80KCY	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LHK	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80M5J	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MBE	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80NLG	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80LFH	SFP+-10G-SR
FPC 14	REV 32	750-028467	ABBN6790	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBK6515	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LZM	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80MCC	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80KCM	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AK80KE0	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+

Xcvr 0	REV 01	740-021310	C10F99155	SFP+-10G-LRM
Xcvr 1	REV 01	740-021310	C10F99049	SFP+-10G-LRM
Xcvr 2	REV 01	740-021310	C10F99128	SFP+-10G-LRM
Xcvr 3	REV 01	740-021310	C10F99169	SFP+-10G-LRM
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E04389	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11F01764	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01879	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01705	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	AK80LEX	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AK80FEU	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AK80FNM	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AJQQQ5G	SFP+-10G-SR
FPC 15	REV 32	750-028467	ABBN6791	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN7289	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00424	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K01849	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01862	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K01852	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00427	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K00430	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01854	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K00426	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	B11K00429	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	B11K01864	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	B11K01850	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	B11K00522	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E01144	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E00985	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E00796	SFP+-10G-USR
Xcvr 3	REV 01	740-031980	B11K01866	SFP+-10G-SR
FPC 16	REV 30	750-028467	ABBM4592	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN0465	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01435	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E01052	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01328	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11F01254	SFP+-10G-USR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02738	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02881	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11F01624	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E00889	SFP+-10G-USR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11E02883	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E00681	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E04306	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E02813	SFP+-10G-USR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-030658	B11F01801	SFP+-10G-USR
Xcvr 1	REV 01	740-030658	B11E02753	SFP+-10G-USR
Xcvr 2	REV 01	740-030658	B11E01156	SFP+-10G-USR
Xcvr 3	REV 01	740-030658	B11E04324	SFP+-10G-USR
FPC 17	REV 32	750-028467	ABBN6810	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN7237	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+

Xcvr 0	REV 01	740-031980	163363A02638	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02082	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A01674	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03058	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A03048	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02729	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02566	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02567	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02878	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02739	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A01959	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02660	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02731	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02588	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02673	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02654	SFP+-10G-SR
FPC 18	REV 30	750-028467	ABBM4739	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBN0487	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02569	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02886	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A03082	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	133363A00297	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02726	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A03050	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02884	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03076	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02581	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02873	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02582	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03083	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031981	UL70BU6	SFP+-10G-LR
Xcvr 1	REV 01	740-031981	UL50QC6	SFP+-10G-LR
Xcvr 2	REV 01	740-031981	UL708N6	SFP+-10G-LR
Xcvr 3	REV 01	740-031981	UL603KK	SFP+-10G-LR
FPC 19	REV 32	750-028467	ABBN6827	MPC 3D 16x 10GE
CPU	REV 10	711-029089	ABBK6508	AMPC PMB
PIC 0		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A01688	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A01724	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A01773	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02593	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A03061	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A03056	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02669	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03070	SFP+-10G-SR
PIC 2		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02572	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02697	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	163363A02585	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A03052	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	4x 10GE(LAN) SFP+
Xcvr 0	REV 01	740-031980	163363A02591	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	163363A02649	SFP+-10G-SR

Xcvr 2	REV 01	740-031980	163363A02577	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	163363A02698	SFP+-10G-SR
ADC 0	REV 07	750-043596	ABBV7229	Adapter Card
ADC 1	REV 11	750-043596	ABBW5689	Adapter Card
ADC 2	REV 05	750-043596	ZZ6196	Adapter Card
ADC 3	REV 07	750-043596	ABBV7242	Adapter Card
ADC 4	REV 07	750-043596	ABBV7227	Adapter Card
ADC 5	REV 07	750-043596	ABBV7207	Adapter Card
ADC 6	REV 07	750-043596	ABBV7218	Adapter Card
ADC 7	REV 07	750-043596	ABBV7200	Adapter Card
ADC 8	REV 11	750-043596	ABBW5688	Adapter Card
ADC 9	REV 07	750-043596	ABBV7235	Adapter Card
ADC 10	REV 07	750-043596	ABBV7198	Adapter Card
ADC 11	REV 06	750-043596	ABBT3805	Adapter Card
ADC 12	REV 07	750-043596	ABBV7180	Adapter Card
ADC 13	REV 07	750-043596	ABBV7187	Adapter Card
ADC 14	REV 11	750-043596	ABBW5671	Adapter Card
ADC 15	REV 11	750-043596	ABBW5674	Adapter Card
ADC 16	REV 11	750-043596	ABBW5673	Adapter Card
ADC 17	REV 11	750-043596	ABBW5678	Adapter Card
ADC 18	REV 11	750-043596	ABBW5682	Adapter Card
ADC 19	REV 11	750-043596	ABBW5685	Adapter Card
Fan Tray 0	REV 2A	760-046960	ACAY0033	172mm FanTray - 6 Fans
Fan Tray 1	REV 2A	760-046960	ACAY0030	172mm FanTray - 6 Fans
Fan Tray 2	REV 2A	760-046960	ACAY0039	172mm FanTray - 6 Fans
Fan Tray 3	REV 2A	760-046960	ACAY0062	172mm FanTray - 6 Fans

To verify Routing Engine status, use the **show chassis routing-engine** command. Sample command output from an MX2020 router is as follows:

```
user@host> show chassis routing-engine
```

Routing Engine status:

Slot 0:

```

Current state           Master
Election priority       Master (default)
Temperature             21 degrees C / 69 degrees F
CPU temperature         20 degrees C / 68 degrees F
DRAM                   17152 MB
Memory utilization      14 percent
CPU utilization:
  User                  2 percent
  Background            0 percent
  Kernel                6 percent
  Interrupt              4 percent
  Idle                  89 percent
Model                  RE-S-1800x4
Serial ID               9009089704
Start time              2012-09-07 13:03:53 PDT
Uptime                  3 days, 11 hours, 50 minutes, 10 seconds
Last reboot reason      Router rebooted after a normal shutdown.
Load averages:          1 minute   5 minute  15 minute
                        0.12       0.05    0.01

```

Routing Engine status:

Slot 1:

```

Current state           Backup
Election priority       Backup (default)
Temperature             21 degrees C / 69 degrees F

```

```

CPU temperature      20 degrees C / 68 degrees F
DRAM                17152 MB
Memory utilization   11 percent
CPU utilization:
  User              0 percent
  Background        0 percent
  Kernel            0 percent
  Interrupt         0 percent
  Idle              100 percent
Model               RE-S-1800x4
Serial ID           9009094138
Start time          2012-09-06 14:52:44 PDT
Uptime              4 days, 10 hours, 1 minute, 21 seconds
Last reboot reason   Router rebooted after a normal shutdown.
Load averages:      1 minute  5 minute  15 minute
                    0.00      0.00      0.00

```

To verify FPC status, use the **show chassis fpc** command. Sample command output from an MX2020 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 29 14 0 2048 18 13
1 Online 26 10 0 2048 18 13
2 Online 25 10 0 2048 18 13
3 Online 26 10 0 2048 18 13
4 Online 27 9 0 2048 18 13
5 Online 27 9 0 2048 18 13
6 Online 27 10 0 2048 18 13
7 Online 27 9 0 2048 18 13
8 Online 27 10 0 2048 18 13
9 Online 28 9 0 2048 18 13
10 Online 32 10 0 2048 18 13
11 Online 28 10 0 2048 18 13
12 Online 27 10 0 2048 18 13
13 Online 26 10 0 2048 18 13
14 Online 28 9 0 2048 18 13
15 Online 28 10 0 2048 18 13
16 Online 28 9 0 2048 18 13
17 Online 27 10 0 2048 18 13
18 Online 28 10 0 2048 18 13
19 Online 30 10 0 2048 18 13

```

Table 90 on page 324 provides an example of **jnxFruEntry** objects in the **jnxFruTable** of a T640 router.

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router

Index	I	Z	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	power EntryModule	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	power EntryModule	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	217044

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (*continued*)

Index	L	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	1	1	0	Top left front fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Top left middle fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top left rear fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Top right front fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	5	0	Top right middle fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	1	6	0	Top right rear fan	fan	1	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Bottom left front fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Bottom left middle fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Bottom left rear fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Bottom right front fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	2	5	0	Bottom right middle fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
4	2	6	0	Bottom right rear fan	fan	2	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	1	0	Fourth blower from top	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	2	0	Bottom blower	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	3	0	Middle blower	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	4	0	Top blower	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
4	3	5	0	Second blower from top	fan	3	Present	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePic Concentrator	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
7	1	1	0	FPC @ 0/0/* top temperature 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 temperature 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 temperature sensor	flexiblePic Concentrator	2	Online	30	None	0:0:00:01.94	0:0:00:00.00	0

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (*continued*)

Index	L	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
7	3	1	0	FPC @ 2/0/* top temper- ature 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 temper- ature 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 temper- ature 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 temper- ature 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 temper- ature sensor	flexiblePic Concentrator	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (*continued*)

Index	L	2	8	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	5	2	0	FPC @ 4/1/* bottom temper- ature sensor	flexiblePic Concentrator	5	Online	36	None	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePic Concentrator	6	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
7	6	1	0	FPC @ 5/0/* top temper- ature 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 temper- ature 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 temper- ature 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 temper- ature 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 temper- ature sensor	flexiblePic Concentrator	8	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	3	4	0	PIC: @ 2/3/*	portInterface- Card	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterface- Card	4	Online		None	0:0:00:01.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterface- Card	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterface- Card	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterface- Card	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	5	1	0	PIC: 1x Tunnel @ 4/0/*	portInterface- Card	5	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
8	5	2	0	PIC: 1x OC-192 SM SR2 @ 4/1/*	portInterface- Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	5	3	0	PIC: 4x OC-48 SONET, SMSR @ 4/2/*	portInterface- Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	5	4	0	PIC: 1x OC-192 SM SR1 @ 4/3/*	portInterface- Card	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterface- Card	6	Empty	0	None	0:0:00:01.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterface- Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterface- Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterface- Card	6	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterface- Card	7	Online	30	None	0:0:00:00.00	0:0:00:00.00	0

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (continued)

Index	L	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	7	2	0	PIC: @ 6/1/*	portInterface- Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterface- Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterface- Card	7	Online	30	None	0:0:00:01.98	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterface- Card	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterface- Card	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterface- Card	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterface- Card	8	Offline	0	Error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routing- Engine	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routing- Engine	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM GBUS	frontPanel- Module	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanel- Module	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	clockGener- ator	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	clockGener- ator	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	control- Board	1	Online	27	None	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	control- Board	2	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

Table 90: jnxFruEntry Objects in the jnxFruTable of a T640 Router (continued)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
14	1	0	0	SPMB 0	processor Mezzanine BoardForSIB	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0
14	2	0	0	SPMB 1	processor Mezzanine BoardForSIB	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	1	0	0	SIB 0	switchInter- faceBoard	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	switchInter- faceBoard	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	switchInter- faceBoard	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	switchInter- faceBoard	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	switchInter- faceBoard	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a T640 router is as follows:

```

user@host> show chassis hardware
Hardware inventory:
Item                Version  Part number  Serial number  Description
Chassis
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
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 91 on page 334](#) provides an example of **jnxFruEntry** objects in the **jnxFruTable** of a T4000 router.

Table 91: 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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	L	P	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
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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	I	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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/*/* MMBO	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 sensor	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 sensor	flexiblePic Concentrator	1	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
7	2	3	0	ST-MMB2 @ 1/*/* MMBO	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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	L	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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/*/* MMBO	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/*/* MMBO	flexiblePic Concentrator	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
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/*/* MMBO	flexiblePic Concentrator	4	Online	36	None	0:0:00:01.98	0:0:00:00.00	0

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (continued)

Index	I	L	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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/*/* MMBO	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
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/*/* MMBO	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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	I	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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/*/* MMBO	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
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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	L	2	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
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

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	L	P	B	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
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
9	2	0	0	Routing Engine 1	routing- Engine	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM GBUS	frontPanel- Module	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanel- Module	1	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	clockGener- ator	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	clockGener- ator	2	Online	27	None	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	control- Board	1	Online	27	None	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	control- Board	2	Unknown	0	None	0:0:00:01.96	0:0:00:00.00	0
13	1	0	0	CIP	craftInter- facePanel	1	Present	36	None	0:0:00:00.00	0:0:00:00.00	0
14	1	0	0	SPMB 0	processor Mezzanine BoardForSIB	1	Online	34	None	0:0:00:00.00	0:0:00:00.00	0

Table 91: jnxFruEntry Objects in the jnxFruTable of a T4000 Router (*continued*)

Index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
14	2	0	0	SPMB 1	processor Mezzanine BoardForSIB	2	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	1	0	0	SIB 0	switchInter- faceBoard	1	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	switchInter- faceBoard	2	Online	36	None	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	switchInter- faceBoard	3	Empty	0	None	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	switchInter- faceBoard	4	Online	30	None	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	switchInter- faceBoard	5	Online	30	None	0:0:00:01.96	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the **show chassis hardware** command. Sample command output from a 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) SFP
  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
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
```

Slot	State	Temp (C)	CPU Utilization (%) Total Interrupt	Memory DRAM (MB)	Utilization (%) 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-DU0-1800
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-DU0-1800
  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:

```

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

```

jnxFruMib

The object identifier for the jnxFruMib is {jnxFruMibRoot 1}. This object shows the status of field replaceable units in an Optical Transport Network(OTN).

- jnxFruCfg— The configuration associated with the FRU, whose object identifier is (jnxFruMib 1)
 - jnxFruCfgTable

The object identifier for the `jnxFruCfgTable` is `{jnxFruCfg 1}`. The object shows the status of the configuration associated with the field-replaceable units (FRUs) in the OTN.

Entries within the `jnxFruCfgTable` are represented by the `jnxFruCfgEntry` object, whose object identifier is `{jnxFruCfgEntry 1}`. The `jnxFruCfgEntry` object contains the following objects, which describe the FRUs in the OTN.

- `jnxFruCfgContentsIndex (1)`— The index value of an entry in `jnxFruCfgEntry`, whose object identifier is `{jnxFruCfgEntry 1}`
- `jnxFruCfgL1Index (2)`— The level-one index associated with the FRU configuration table, whose object identifier is `{jnxFruCfgEntry 2}`
- `jnxFruCfgL2Index (3)`— The level-two index associated with the FRU configuration table, whose object identifier is `{jnxFruCfgEntry 3}`
- `jnxFruCfgL3Index (4)`— The level-three index associated with the FRU configuration table, whose object identifier is `{jnxFruCfgEntry 4}`
- `jnxFruCfgType (5)`— The FRU type associated with the FRU configuration table, whose object identifier is `{jnxFruCfgEntry 5}`
- `jnxFruCfgAdminState (6)`— The administration state of the FRU configuration table, whose object identifier is `{jnxFruCfgEntry 6}`
- `jnxFruCfgOperState (7)`— The operational state of the set and get operations in FRU configuration table, whose object identifier is `{jnxFruCfgEntry 7}`

`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.

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 57](#).

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 a six-input DC power supply, you must also inspect the **feeds** configuration. For the six-input DC power supply, the **jnxPowerSupplyFailure** trap is sent when the number of input feeds connected to the power supply is less than the configured value.



NOTE:

1. The number of input feeds can be configured by using the **set chassis pem feed number-of-input-feeds** statement at the [edit chassis pem] or [edit chassis lcc lcc-number pem] hierarchy level.
2. The behavior stated here for the six-input DC power supply is applicable only on T640, T6100, and T4000 platforms.
3. For a two-input DC power supply, no SNMP trap is generated when either feed fails.

- **Fan failure (jnxFanFailure)**—Sent when the fan fuse blows or when the fan wiring shorts out. When only one fan has failed, there is no service impact. The remaining fans increase speed to compensate. However, you must resolve the problem before another fan fails. This trap is repeated every hour until the fan failure is fixed. To determine the source of the failure, you must physically inspect the router, taking care to check the fuses. See the hardware installation guide for your router model for more information.
- **Overtemperature (jnxOverTemperature)**—Sent when several fans fail or the room temperature increases significantly. The service impact of this trap depends on the temperature of the router. In general, the router increases the speed of the fans when any component exceeds a temperature of 55° C. The fans remain at the higher speed until the temperature decreases below the threshold. In this case, there is no service impact. However, if the temperature exceeds 75° C, the router transmits a warning and automatically shuts down. This scenario creates a significant service impact because the shutdown affects additional routers and equipment. This trap is repeated every minute until the temperature is brought down to normal. To determine the source of the overtemperature problem, you must physically inspect the router to determine whether any fans have failed in the router.
- **Power Supply OK (jnxPowerSupplyOK)**—Sent when a power supply recovers from failure. (The six-input DC power supply fails when the number of input feeds connected to the power supply is less than the configured value.)

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.
- 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 349](#)
- [SNMPv2 Trap Format on page 350](#)

SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in [Table 92 on page 349](#). To view the SNMPv1 chassis-related traps, see “[Juniper Networks Enterprise-Specific SNMP Version 1 Traps](#)” on page 894 .

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 92: 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

Table 92: SNMP Version 1 Trap Format (*continued*)

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2
jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12
jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9
jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6
jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10
jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8
jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5
jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2	6	1
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1	6	4
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3

SNMPv2 Trap Format

The SNMPv2 trap format for the Chassis MIB traps is described in [Table 93 on page 351](#).

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 93: 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.
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.

Table 93: SNMP Version 2 Trap Format (*continued*)

Trap Name	snmpTrapOID	Description
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	The component sensor has detected an overtemperature condition.

Chassis Definitions for Router Model MIB

The enterprise-specific Chassis Definitions for Router Model MIB contain the OIDs that are used by the Chassis MIB to identify platform and chassis components. The Chassis MIB provides information that changes often. The Chassis Definitions for Router Model MIB provide information that changes less often.

The last number in each **sysObjectId**, shown in [Table 94 on page 352](#), corresponds to the router model and therefore does not change.

Table 94: Router Models and Their sysObjectIds

Model	SysObjectId	jnxProductName
J2300	1.3.6.1.4.1.2636.1.1.1.2.13	jnxProductNameJ2300
J4300	1.3.6.1.4.1.2636.1.1.1.2.14	jnxProductNameJ4300
J6300	1.3.6.1.4.1.2636.1.1.1.2.15	jnxProductNameJ6300
M5	1.3.6.1.4.1.2636.1.1.1.2.5	jnxProductNameM5
M7i	1.3.6.1.4.1.2636.1.1.1.2.10	jnxProductNameM7i
M10	1.3.6.1.4.1.2636.1.1.1.2.4	jnxProductNameM10
M10i	1.3.6.1.4.1.2636.1.1.1.2.11	jnxProductNameM10i
M20	1.3.6.1.4.1.2636.1.1.1.2.2	jnxProductNameM20
M40	1.3.6.1.4.1.2636.1.1.1.2.1	jnxProductNameM40
M40e	1.3.6.1.4.1.2636.1.1.1.2.8	jnxProductNameM40e
M120	1.3.6.1.4.1.2636.1.1.1.2.18	jnxProductNameM120
M160	1.3.6.1.4.1.2636.1.1.1.2.3	jnxProductNameM160
M320	1.3.6.1.4.1.2636.1.1.1.2.9	jnxProductNameM320
MX2020	1.3.6.1.4.1.2636.1.1.1.2.93	jnxProductNameMX2020
MX2010	1.3.6.1.4.1.2636.1.1.1.2.99	jnxProductNameMX2010

Table 94: Router Models and Their sysObjectIds (*continued*)

Model	SysObjectID	jnxProductName
MX960	1.3.6.1.4.1.2636.1.1.1.2.21	jnxProductNameMX960
MX480	1.3.6.1.4.1.2636.1.1.1.1.25	jnxProductNameMX480
MX240	1.3.6.1.4.1.2636.1.1.1.1.29	jnxProductNameMX240
MX80	1.3.6.1.4.1.2636.1.1.1.1.57	jnxProductNameMX80
MX40	1.3.6.1.4.1.2636.1.1.1.1.88	jnxProductNameMX40
MX10	1.3.6.1.4.1.2636.1.1.1.1.89	jnxProductNameMX10
MX5	1.3.6.1.4.1.2636.1.1.1.1.90	jnxProductNameMX5
EX2200	1.3.6.1.4.1.2636.1.1.1.1.43	jnxProductNameEX2200
EX3200	1.3.6.1.4.1.2636.1.1.1.2.30	jnxProductNameEX3200
EX4200	1.3.6.1.4.1.2636.1.1.1.1.31	jnxProductNameEX4200
EX4500	1.3.6.1.4.1.2636.1.1.1.1.44	jnxProductNameEX4500
EX8208	1.3.6.1.4.1.2636.1.1.1.1.32	jnxProductNameEX8208
EX8216	1.3.6.1.4.1.2636.1.1.1.1.33	jnxProductNameEX8216
SRX100	1.3.6.1.4.1.2636.1.1.1.2.41	jnxProductNameSRX100
SRX210	1.3.6.1.4.1.2636.1.1.1.2.36	jnxProductNameSRX210
SRX220	1.3.6.1.4.1.2636.1.1.1.2.58	jnxProductNameSRX220
SRX240	1.3.6.1.4.1.2636.1.1.1.2.39	jnxProductNameSRX240
SRX550	1.3.6.1.4.1.2636.1.1.1.2.86	jnxProductNameSRX550
SRX650	1.3.6.1.4.1.2636.1.1.1.2.40	jnxProductNameSRX650
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
SRX5600	1.3.6.1.4.1.2636.1.1.1.2.28	jnxProductNameSRX5600
SRX5800	1.3.6.1.4.1.2636.1.1.1.2.26	jnxProductNameSRX5800

Table 94: Router Models and Their sysObjectIds (*continued*)

Model	SysObjectId	jnxProductName
TX	1.3.6.1.4.1.2636.1.1.1.2.17	jnxProductNameTX
T320	1.3.6.1.4.1.2636.1.1.1.2.7	jnxProductNameT320
T640	1.3.6.1.4.1.2636.1.1.1.2.6	jnxProductNameT640
T1600	1.3.6.1.4.1.2636.1.1.1.2.27	jnxProductNameT1600
T4000	1.3.6.1.4.1.2636.1.1.1.2.83	jnxProductNameT4000
TX Plus	1.3.6.1.4.1.2636.1.1.1.2.37	jnxProductNameTXP

For a downloadable version of the Chassis Definitions for Router Model MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chas-defines.txt.

MIB Objects for the M120 Router

New Chassis MIB objects for the M120 router include:

```

jnxProductLineM120      OBJECT IDENTIFIER ::= { jnxProductLine      18 }
jnxProductNameM120      OBJECT IDENTIFIER ::= { jnxProductName      18 }
jnxProductModelM120     OBJECT IDENTIFIER ::= { jnxProductModel     18 }
jnxProductVariationM120 OBJECT IDENTIFIER ::= { jnxProductVariation 18 }
jnxChassisM120          OBJECT IDENTIFIER ::= { jnxChassis          18 }
jnxSlotM120             OBJECT IDENTIFIER ::= { jnxSlot             18 }
    jnxM120SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotM120 1 }
    jnxM120SlotFEB       OBJECT IDENTIFIER ::= { jnxSlotM120 2 }
    jnxM120SlotHM        OBJECT IDENTIFIER ::= { jnxSlotM120 3 }
    jnxM120SlotPower     OBJECT IDENTIFIER ::= { jnxSlotM120 4 }
    jnxM120SlotFan       OBJECT IDENTIFIER ::= { jnxSlotM120 5 }
    jnxM120SlotCB        OBJECT IDENTIFIER ::= { jnxSlotM120 6 }
    jnxM120SlotFPB       OBJECT IDENTIFIER ::= { jnxSlotM120 7 }

jnxMediaCardSpaceM120   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   18 }
jnxM120MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceM120 1 }

jnxMidplaneM120         OBJECT IDENTIFIER ::= { jnxBackplane        18 }
jnxModuleM120           OBJECT IDENTIFIER ::= { jnxModule          18 }
    jnxM120FEB           OBJECT IDENTIFIER ::= { jnxModuleM120 1 }

```



NOTE: The M120 router does not support the enterprise-specific Dynamic Flow Capture MIB.

Sample command output from the **show chassis hardware** command for the M120 router is as follows:

```

user@host> show chassis hardware
Hardware inventory:
Item                Version  Part number  Serial number  Description

```

Chassis			JN000019AC	M120
Midplane	REV 01	710-011382	RB3003	M120 Midplane
FPM Board	REV 01	710-011407	CK9165	M120 FPM Board
FPM Display	REV 01	710-011405	CE0032	M120 FPM Display
FPM CIP	REV 01	710-011410	CE0058	M120 FPM CIP
PEM 1	Rev 01	740-011935	RG10165	DC Power Entry Module
Routing Engine 0	REV 00	740-014082	1000604605	RE-A-2000
Routing Engine 1	REV 00	740-014082	1000604601	RE-A-2000
CB 0	REV 03	710-011403	CM8335	M120 Control Board
CB 1	REV 03	710-011403	CM8340	M120 Control Board
FPC 0	REV 01	710-012879	CH1622	M120 CFPC OC192
PIC 0		BUILTIN	BUILTIN	1x OC-192 SONET XFP
Xcvr 0		NON-JNPR	T05J32698	XFP-OC192-SR
FPC 1	REV 01	710-012882	CE0062	M120 CFPC 10GE
PIC 0		BUILTIN	BUILTIN	1x 10GE(LAN/WAN) XFP
Xcvr 0		NON-JNPR	T05A02227	XFP-10G-ER
FPC 2	REV 01	710-011388	CJ9092	M120 FPC Type 1
PIC 0	REV 16	750-007444	HS1526	1x CHOC3 IQ SONET, SMIR
PIC 1	REV 12	750-005637	HT0533	4x CHDS3 IQ
PIC 2	REV 15	750-005634	HN1903	1x CHOC12 IQ SONET, SMIR
PIC 3	REV 15	750-007631	NB5006	10x CHE1 IQ
Board B	REV 01	710-011390	CJ9109	M120 FPC Mezz Board
FPC 3	REV 03	710-011393	CJ9231	M120 FPC Type 2
PIC 0	REV 05	750-010472	JE3146	1x OC-48 ATM-II IQ
Xcvr 0	REV 01	740-009028	P5F05WU	SFP-SR
PIC 1	REV 13	750-001901	HB4231	4x OC-12 SONET, SMIR
PIC 2	REV 15	750-008155	HX5442	2x G/E IQ, 1000 BASE
Xcvr 0	REV	740-007326	P11E5RR	SFP-SX
Xcvr 1	REV 01	740-009029	4C81050	UNKNOWN
PIC 3	REV 16	750-008155	HZ8871	2x G/E IQ, 1000 BASE
Xcvr 0	REV 01	740-011613	P8E2KGF	SFP-SX
Xcvr 1	REV 01	740-011782	P6M1E5X	SFP-SX
Board B	REV 02	710-011395	CN3750	M120 FPC Mezz
FPC 4	REV 01	710-011388	CJ9089	M120 FPC Type 1
PIC 0	REV 03	750-002911	AJ2279	4x F/E, 100 BASE-TX
PIC 1	REV 15	750-005634	HN0435	1x CHOC12 IQ SONET, SMIR
PIC 2	REV 02	750-003064	HD4548	4x T1, RJ48
PIC 3	REV 04	750-011209	JC8254	Adaptive Services-II
Board B	REV 01	710-011390	CJ9111	M120 FPC Mezz Board
FPC 5	REV 01	710-011388	CJ9360	M120 FPC Type 1
PIC 0	REV 08	750-007631	HK0212	10x CHE1 IQ
PIC 1	REV 05	750-003034	BD8705	4x OC-3 SONET, SMIR
PIC 2	REV 11	750-007643	NA5967	1x G/E IQ, 1000 BASE
Xcvr 0	REV 01	740-007326	P4R0PNZ	SFP-SX
PIC 3	REV 16	750-007444	HS1501	1x CHSTM1 IQ SDH, SMIR
Board B	REV 01	710-011390	CJ9099	M120 FPC Mezz Board
FEB 0	REV 04	710-011663	CJ9364	M120 FEB
FEB 1	REV 04	710-011663	CJ9385	M120 FEB
FEB 2	REV 02	710-015795	CP6830	M120 FEB
FEB 3	REV 01	710-011663	CM2585	M120 FEB
FEB 4	REV 04	710-011663	CJ9416	M120 FEB
FEB 5	REV 01	710-011663	CM2600	M120 FEB

MIB Objects for the MX2020 3D Universal Edge Router

The Chassis MIB objects for the MX2020 3D Universal Edge Router include:

```

jnxProductLineMX2020      OBJECT IDENTIFIER ::= { jnxProductLine      93 }
jnxProductNameMX2020      OBJECT IDENTIFIER ::= { jnxProductName      93 }
jnxProductModelMX2020     OBJECT IDENTIFIER ::= { jnxProductModel     93 }
jnxProductVariationMX2020 OBJECT IDENTIFIER ::= { jnxProductVariation 93 }
jnxChassisMX2020          OBJECT IDENTIFIER ::= { jnxChassis          93 }
jnxSlotMX2020             OBJECT IDENTIFIER ::= { jnxSlot             93 }
    jnxMX2020SlotSFB       OBJECT IDENTIFIER ::= { jnxSlotMX2020 1 }
        -- Switch Fabric Board
    jnxMX2020SlotHM        OBJECT IDENTIFIER ::= { jnxSlotMX2020 2 }
        -- Host Module (also called Routing Engine {RE})
    jnxMX2020SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotMX2020 3 }
        -- Flexible Port Concentrator slot
    jnxMX2020SlotFan       OBJECT IDENTIFIER ::= { jnxSlotMX2020 4 }
    jnxMX2020SlotCB        OBJECT IDENTIFIER ::= { jnxSlotMX2020 5 }
        -- Control Board (hosts RE, SPMB)
    jnxMX2020SlotFPB       OBJECT IDENTIFIER ::= { jnxSlotMX2020 6 }
        -- Front Panel Board
    jnxMX2020SlotSPMB      OBJECT IDENTIFIER ::= { jnxSlotMX2020 7 }
        -- Processor Mezzanine Board for SFB
    jnxMX2020SlotPDM       OBJECT IDENTIFIER ::= { jnxSlotMX2020 8 }
        -- Power Distribution Module
    jnxMX2020SlotPSM       OBJECT IDENTIFIER ::= { jnxSlotMX2020 9 }
        -- Power Supply Module
    jnxMX2020SlotADC       OBJECT IDENTIFIER ::= { jnxSlotMX2020 10 }
        -- Adapter Card (connects FPC to backplane)

jnxMediaCardSpaceMX2020   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   93 }
    jnxMX2020MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX2020 1
}
    jnxMX2020MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX2020 2
}

jnxBackplaneMX2020        OBJECT IDENTIFIER ::= { jnxBackplane        93 }
    jnxBackplaneLowerMX2020 OBJECT IDENTIFIER ::= { jnxBackplaneMX2020
1 }
    jnxBackplaneUpperMX2020 OBJECT IDENTIFIER ::= { jnxBackplaneMX2020
2 }
    jnxBackplaneLowerPowerMX2020 OBJECT IDENTIFIER ::= { jnxBackplaneMX2020
3 }
    jnxBackplaneUpperPowerMX2020 OBJECT IDENTIFIER ::= { jnxBackplaneMX2020
4 }

jnxModuleMX2020           OBJECT IDENTIFIER ::= { jnxModule           93 }
    jnxMX2020SFB           OBJECT IDENTIFIER ::= { jnxModuleMX2020 1 }
    jnxMX2020HM            OBJECT IDENTIFIER ::= { jnxModuleMX2020 2 }
    jnxMX2020FPC           OBJECT IDENTIFIER ::= { jnxModuleMX2020 3 }
    jnxMX2020Fan           OBJECT IDENTIFIER ::= { jnxModuleMX2020 4 }
    jnxMX2020CB            OBJECT IDENTIFIER ::= { jnxModuleMX2020 5 }
    jnxMX2020FPB           OBJECT IDENTIFIER ::= { jnxModuleMX2020 6 }
    jnxMX2020SPMB          OBJECT IDENTIFIER ::= { jnxModuleMX2020 7 }
    jnxMX2020PDM           OBJECT IDENTIFIER ::= { jnxModuleMX2020 8 }
    jnxMX2020PSM           OBJECT IDENTIFIER ::= { jnxModuleMX2020 9 }
    jnxMX2020ADC           OBJECT IDENTIFIER ::= { jnxModuleMX2020 10 }

```

MIB Objects for the MX2010 3D Universal Edge Router

The Chassis MIB objects for the MX2010 3D Universal Edge Router include:

```

jnxProductLineMX2010      OBJECT IDENTIFIER ::= { jnxProductLine      99 }
jnxProductNameMX2010      OBJECT IDENTIFIER ::= { jnxProductName      99 }
jnxProductModelMX2010     OBJECT IDENTIFIER ::= { jnxProductModel     99 }
jnxProductVariationMX2010 OBJECT IDENTIFIER ::= { jnxProductVariation 99 }
jnxChassisMX2010          OBJECT IDENTIFIER ::= { jnxChassis          99 }

jnxSlotMX2010             OBJECT IDENTIFIER ::= { jnxSlot             99 }
  jnxMX2010SlotSFB         OBJECT IDENTIFIER ::= { jnxSlotMX2010 1 }
                           -- Switch Fabric Board
  jnxMX2010SlotHM          OBJECT IDENTIFIER ::= { jnxSlotMX2010 2 }
                           -- Host Module (also called Routing Engine {RE})
  jnxMX2010SlotFPC         OBJECT IDENTIFIER ::= { jnxSlotMX2010 3 }
                           -- Flexible Port Concentrator slot
  jnxMX2010SlotFan         OBJECT IDENTIFIER ::= { jnxSlotMX2010 4 }
  jnxMX2010SlotCB          OBJECT IDENTIFIER ::= { jnxSlotMX2010 5 }
                           -- Control Board (hosts RE, SPMB)
  jnxMX2010SlotFPB         OBJECT IDENTIFIER ::= { jnxSlotMX2010 6 }
                           -- Front Panel Board
  jnxMX2010SlotSPMB        OBJECT IDENTIFIER ::= { jnxSlotMX2010 7 }
                           -- Processor Mezzanine Board for SFB
  jnxMX2010SlotPDM         OBJECT IDENTIFIER ::= { jnxSlotMX2010 8 }
                           -- Power Distribution Module
  jnxMX2010SlotPSM         OBJECT IDENTIFIER ::= { jnxSlotMX2010 9 }
                           -- Power Supply Module
  jnxMX2010SlotADC         OBJECT IDENTIFIER ::= { jnxSlotMX2010 10 }
                           -- Adapter Card (connects FPC to backplane)

jnxMediaCardSpaceMX2010   OBJECT IDENTIFIER ::= { jnxMediaCardSpace   99 }
  jnxMX2010MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX2010 1
}
  jnxMX2010MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX2010 2
}

jnxBackplaneMX2010        OBJECT IDENTIFIER ::= { jnxBackplane        99 }
  jnxBackplaneLowerMX2010  OBJECT IDENTIFIER ::= { jnxBackplaneMX2010 1 }
  jnxBackplaneUpperMX2010  OBJECT IDENTIFIER ::= { jnxBackplaneMX2010 2 }
  jnxBackplanePowerMX2010  OBJECT IDENTIFIER ::= { jnxBackplaneMX2010 3 }

jnxModuleMX2010           OBJECT IDENTIFIER ::= { jnxModule           99 }
  jnxMX2010SFB             OBJECT IDENTIFIER ::= { jnxModuleMX2010 1 }
  jnxMX2010HM              OBJECT IDENTIFIER ::= { jnxModuleMX2010 2 }
  jnxMX2010FPC             OBJECT IDENTIFIER ::= { jnxModuleMX2010 3 }
  jnxMX2010Fan             OBJECT IDENTIFIER ::= { jnxModuleMX2010 4 }
  jnxMX2010CB              OBJECT IDENTIFIER ::= { jnxModuleMX2010 5 }
  jnxMX2010FPB             OBJECT IDENTIFIER ::= { jnxModuleMX2010 6 }
  jnxMX2010SPMB            OBJECT IDENTIFIER ::= { jnxModuleMX2010 7 }
  jnxMX2010PDM             OBJECT IDENTIFIER ::= { jnxModuleMX2010 8 }
  jnxMX2010PSM            OBJECT IDENTIFIER ::= { jnxModuleMX2010 9 }
  jnxMX2010ADC             OBJECT IDENTIFIER ::= { jnxModuleMX2010 10 }

```

MIB Objects for the MX960 3D Universal Edge Router

The Chassis MIB objects for the MX960 3D Universal Edge Router include:

```
jnxProductLineX960      OBJECT IDENTIFIER ::= { jnxProductLine      21 }
jnxProductNameX960      OBJECT IDENTIFIER ::= { jnxProductName      21 }
jnxProductModelX960     OBJECT IDENTIFIER ::= { jnxProductModel    21 }
jnxProductVariationX960 OBJECT IDENTIFIER ::= { jnxProductVariation 21 }
jnxChassisX960          OBJECT IDENTIFIER ::= { jnxChassis        21 }
jnxSlotX960             OBJECT IDENTIFIER ::= { jnxSlot           21 }
  jnxX960SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotX960 1 }
  jnxX960SlotHM         OBJECT IDENTIFIER ::= { jnxSlotX960 2 }
  jnxX960SlotPower      OBJECT IDENTIFIER ::= { jnxSlotX960 3 }
  jnxX960SlotFan        OBJECT IDENTIFIER ::= { jnxSlotX960 4 }
  jnxX960SlotCB         OBJECT IDENTIFIER ::= { jnxSlotX960 5 }
  jnxX960SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotX960 6 }
jnxMediaCardSpaceX960   OBJECT IDENTIFIER ::= { jnxMediaCardSpace 21 }
  jnxX960MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceX960 1 }
jnxMidplaneX960        OBJECT IDENTIFIER ::= { jnxBackplane      21 }
```

MIB Objects for the MX480 3D Universal Edge Router

The Chassis MIB objects for the MX480 3D Universal Edge Router include:

```
jnxProductLineMX480     OBJECT IDENTIFIER ::= { jnxProductLine      25 }
jnxProductNameMX480     OBJECT IDENTIFIER ::= { jnxProductName      25 }
jnxProductModelMX480    OBJECT IDENTIFIER ::= { jnxProductModel    25 }
jnxProductVariationMX480 OBJECT IDENTIFIER ::= { jnxProductVariation 25 }
jnxChassisMX480         OBJECT IDENTIFIER ::= { jnxChassis        25 }

jnxSlotMX480            OBJECT IDENTIFIER ::= { jnxSlot           25 }
  jnxMX480SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotMX480 1 }
  jnxMX480SlotHM        OBJECT IDENTIFIER ::= { jnxSlotMX480 2 }
  jnxMX480SlotPower     OBJECT IDENTIFIER ::= { jnxSlotMX480 3 }
  jnxMX480SlotFan       OBJECT IDENTIFIER ::= { jnxSlotMX480 4 }
  jnxMX480SlotCB        OBJECT IDENTIFIER ::= { jnxSlotMX480 5 }
  jnxMX480SlotFPB       OBJECT IDENTIFIER ::= { jnxSlotMX480 6 }

jnxMediaCardSpaceMX480  OBJECT IDENTIFIER ::= { jnxMediaCardSpace 25 }
jnxMX480MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX480 1 }

jnxMidplaneMX480        OBJECT IDENTIFIER ::= { jnxBackplane      25 }
```

MIB Objects for the MX240 3D Universal Edge Router

The Chassis MIB objects for the MX240 3D Universal Edge Router include:

```
jnxProductLineMX240     OBJECT IDENTIFIER ::= { jnxProductLine      29 }
jnxProductNameMX240     OBJECT IDENTIFIER ::= { jnxProductName      29 }
jnxProductModelMX240    OBJECT IDENTIFIER ::= { jnxProductModel    29 }
jnxProductVariationMX240 OBJECT IDENTIFIER ::= { jnxProductVariation 29 }
jnxChassisMX240         OBJECT IDENTIFIER ::= { jnxChassis        29 }

jnxSlotMX240            OBJECT IDENTIFIER ::= { jnxSlot           29 }
  jnxMX240SlotFPC       OBJECT IDENTIFIER ::= { jnxSlotMX240 1 }
  jnxMX240SlotHM        OBJECT IDENTIFIER ::= { jnxSlotMX240 2 }
  jnxMX240SlotPower     OBJECT IDENTIFIER ::= { jnxSlotMX240 3 }
```

```

jnxMX240SlotFan      OBJECT IDENTIFIER ::= { jnxSlotMX240 4 }
jnxMX240SlotCB       OBJECT IDENTIFIER ::= { jnxSlotMX240 5 }
jnxMX240SlotFPB      OBJECT IDENTIFIER ::= { jnxSlotMX240 6 }

jnxMediaCardSpaceMX240 OBJECT IDENTIFIER ::= { jnxMediaCardSpace 29 }
jnxMX240MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX240 1 }

jnxMidplaneMX240     OBJECT IDENTIFIER ::= { jnxBackplane 29 }

```

MIB Objects for the MX80 3D Universal Edge Router

The Chassis MIB objects for the MX80 3D Universal Edge Router include:

```

jnxProductLineMX80    OBJECT IDENTIFIER ::= { jnxProductLine 57 }
jnxProductNameMX80    OBJECT IDENTIFIER ::= { jnxProductName 57 }
jnxProductModelMX80   OBJECT IDENTIFIER ::= { jnxProductModel 57 }
jnxProductVariationMX80 OBJECT IDENTIFIER ::= { jnxProductVariation 57 }
  jnxProductMX80      OBJECT IDENTIFIER ::= { jnxProductVariationMX80 1 }
  jnxProductMX80-48T  OBJECT IDENTIFIER ::= { jnxProductVariationMX80 2 }
  jnxProductMX80-T    OBJECT IDENTIFIER ::= { jnxProductVariationMX80 3 }
jnxChassisMX80        OBJECT IDENTIFIER ::= { jnxChassis 57 }

jnxSlotMX80           OBJECT IDENTIFIER ::= { jnxSlot 57 }
  jnxMX80SlotFPC      OBJECT IDENTIFIER ::= { jnxSlotMX80 1 }
  jnxMX80SlotCFEB     OBJECT IDENTIFIER ::= { jnxSlotMX80 2 }
  jnxMX80SlotRE       OBJECT IDENTIFIER ::= { jnxSlotMX80 3 }
  jnxMX80SlotPower    OBJECT IDENTIFIER ::= { jnxSlotMX80 4 }
  jnxMX80SlotFan      OBJECT IDENTIFIER ::= { jnxSlotMX80 5 }

jnxMediaCardSpaceMX80 OBJECT IDENTIFIER ::= { jnxMediaCardSpace 57 }
  jnxMX80MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX80 1 }
  jnxMX80MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX80 2 }

jnxMidplaneMX80       OBJECT IDENTIFIER ::= { jnxBackplane 57 }

jnxModuleMX80         OBJECT IDENTIFIER ::= { jnxModule 57 }
  jnxMX80FPC          OBJECT IDENTIFIER ::= { jnxModuleMX80 1 }
  jnxMX80CFEB         OBJECT IDENTIFIER ::= { jnxModuleMX80 2 }
  jnxMX80RE           OBJECT IDENTIFIER ::= { jnxModuleMX80 3 }
  jnxMX80Power        OBJECT IDENTIFIER ::= { jnxModuleMX80 4 }
  jnxMX80PowerAC      OBJECT IDENTIFIER ::= { jnxModuleMX80 5 }
  jnxMX80Fan          OBJECT IDENTIFIER ::= { jnxModuleMX80 6 }

```

MIB Objects for the MX40 3D Universal Edge Router

The Chassis MIB objects for the MX40 3D Universal Edge Router include:

```

jnxProductLineMX40    OBJECT IDENTIFIER ::= { jnxProductLine 88 }
jnxProductNameMX40    OBJECT IDENTIFIER ::= { jnxProductName 88 }
jnxProductModelMX40   OBJECT IDENTIFIER ::= { jnxProductModel 88 }
jnxProductVariationMX40 OBJECT IDENTIFIER ::= { jnxProductVariation 88 }
  jnxProductMX40      OBJECT IDENTIFIER ::= { jnxProductVariationMX40 1 }
jnxChassisMX40        OBJECT IDENTIFIER ::= { jnxChassis 88 }

jnxSlotMX40           OBJECT IDENTIFIER ::= { jnxSlot 88 }
  jnxMX40SlotFPC      OBJECT IDENTIFIER ::= { jnxSlotMX40 1 }
  jnxMX40SlotCFEB     OBJECT IDENTIFIER ::= { jnxSlotMX40 2 }

```

```

jnxMX40SlotRE          OBJECT IDENTIFIER ::= { jnxSlotMX40 3 }
jnxMX40SlotPower       OBJECT IDENTIFIER ::= { jnxSlotMX40 4 }
jnxMX40SlotFan         OBJECT IDENTIFIER ::= { jnxSlotMX40 5 }

jnxMediaCardSpaceMX40  OBJECT IDENTIFIER ::= { jnxMediaCardSpace 88 }
jnxMX40MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX40 1 }
jnxMX40MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX40 2 }

jnxMidplaneMX40       OBJECT IDENTIFIER ::= { jnxBackplane 88 }

jnxModuleMX40         OBJECT IDENTIFIER ::= { jnxModule 88 }
jnxMX40FPC            OBJECT IDENTIFIER ::= { jnxModuleMX40 1 }
jnxMX40CFEB          OBJECT IDENTIFIER ::= { jnxModuleMX40 2 }
jnxMX40RE             OBJECT IDENTIFIER ::= { jnxModuleMX40 3 }
jnxMX40Power          OBJECT IDENTIFIER ::= { jnxModuleMX40 4 }
jnxMX40PowerAC        OBJECT IDENTIFIER ::= { jnxModuleMX40 5 }
jnxMX40Fan            OBJECT IDENTIFIER ::= { jnxModuleMX40 6 }

```

MIB Objects for the MX10 3D Universal Edge Router

The Chassis MIB objects for the MX10 3D Universal Edge Router include:

```

jnxProductLineMX10     OBJECT IDENTIFIER ::= { jnxProductLine 89 }
jnxProductNameMX10     OBJECT IDENTIFIER ::= { jnxProductName 89 }
jnxProductModelMX10    OBJECT IDENTIFIER ::= { jnxProductModel 89 }
jnxProductVariationMX10 OBJECT IDENTIFIER ::= { jnxProductVariation 89 }
jnxProductMX10         OBJECT IDENTIFIER ::= { jnxProductVariationMX10 1 }
jnxChassisMX10         OBJECT IDENTIFIER ::= { jnxChassis 89 }

jnxSlotMX10           OBJECT IDENTIFIER ::= { jnxSlot 89 }
jnxMX10SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotMX10 1 }
jnxMX10SlotCFEB       OBJECT IDENTIFIER ::= { jnxSlotMX10 2 }
jnxMX10SlotRE         OBJECT IDENTIFIER ::= { jnxSlotMX10 3 }
jnxMX10SlotPower      OBJECT IDENTIFIER ::= { jnxSlotMX10 4 }
jnxMX10SlotFan        OBJECT IDENTIFIER ::= { jnxSlotMX10 5 }

jnxMediaCardSpaceMX10  OBJECT IDENTIFIER ::= { jnxMediaCardSpace 89 }
jnxMX10MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX10 1 }
jnxMX10MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX10 2 }

jnxMidplaneMX10       OBJECT IDENTIFIER ::= { jnxBackplane 89 }

jnxModuleMX10         OBJECT IDENTIFIER ::= { jnxModule 89 }
jnxMX10FPC            OBJECT IDENTIFIER ::= { jnxModuleMX10 1 }
jnxMX10CFEB          OBJECT IDENTIFIER ::= { jnxModuleMX10 2 }
jnxMX10RE             OBJECT IDENTIFIER ::= { jnxModuleMX10 3 }
jnxMX10Power          OBJECT IDENTIFIER ::= { jnxModuleMX10 4 }
jnxMX10PowerAC        OBJECT IDENTIFIER ::= { jnxModuleMX10 5 }
jnxMX10Fan            OBJECT IDENTIFIER ::= { jnxModuleMX10 6 }

```

MIB Objects for the MX5 3D Universal Edge Router

The Chassis MIB objects for the MX5 3D Universal Edge Router include:

```

jnxProductLineMX5     OBJECT IDENTIFIER ::= { jnxProductLine 90 }
jnxProductNameMX5     OBJECT IDENTIFIER ::= { jnxProductName 90 }
jnxProductModelMX5    OBJECT IDENTIFIER ::= { jnxProductModel 90 }
jnxProductVariationMX5 OBJECT IDENTIFIER ::= { jnxProductVariation 90 }

```



```

jnxProductMX5          OBJECT IDENTIFIER ::= { jnxProductVariationMX5 1 }
jnxChassisMX5          OBJECT IDENTIFIER ::= { jnxChassis          90 }

jnxSlotMX5             OBJECT IDENTIFIER ::= { jnxSlot          90 }
jnxMX5SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotMX5 1 }
jnxMX5SlotCFEB         OBJECT IDENTIFIER ::= { jnxSlotMX5 2 }
jnxMX5SlotRE           OBJECT IDENTIFIER ::= { jnxSlotMX5 3 }
jnxMX5SlotPower        OBJECT IDENTIFIER ::= { jnxSlotMX5 4 }
jnxMX5SlotFan          OBJECT IDENTIFIER ::= { jnxSlotMX5 5 }

jnxMediaCardSpaceMX5   OBJECT IDENTIFIER ::= { jnxMediaCardSpace 90 }
jnxMX5MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX5 1 }
jnxMX5MediaCardSpaceMIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceMX5 2 }

jnxMidplaneMX5         OBJECT IDENTIFIER ::= { jnxBackplane      90 }

jnxModuleMX5           OBJECT IDENTIFIER ::= { jnxModule        90 }
jnxMX5FPC              OBJECT IDENTIFIER ::= { jnxModuleMX5 1 }
jnxMX5CFEB             OBJECT IDENTIFIER ::= { jnxModuleMX5 2 }
jnxMX5RE               OBJECT IDENTIFIER ::= { jnxModuleMX5 3 }
jnxMX5Power            OBJECT IDENTIFIER ::= { jnxModuleMX5 4 }
jnxMX5PowerAC          OBJECT IDENTIFIER ::= { jnxModuleMX5 5 }
jnxMX5Fan              OBJECT IDENTIFIER ::= { jnxModuleMX5 6 }

```

MIB Objects for the PTX5000 Packet Transport Router

The Chassis MIB objects for the PTX5000 Services Gateway include:

```

jnxProductLinePTX5000  OBJECT IDENTIFIER ::= { jnxProductLine    69 }
jnxProductNamePTX5000  OBJECT IDENTIFIER ::= { jnxProductName    69 }
jnxProductModelPTX5000 OBJECT IDENTIFIER ::= { jnxProductModel   69 }
jnxProductVariationPTX5000 OBJECT IDENTIFIER ::= { jnxProductVariation 69 }
jnxChassisPTX5000      OBJECT IDENTIFIER ::= { jnxChassis        69 }

jnxSlotPTX5000         OBJECT IDENTIFIER ::= { jnxSlot          69 }
jnxPTX5000SlotSIB      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 1 }
jnxPTX5000SlotHM       OBJECT IDENTIFIER ::= { jnxSlotPTX5000 2 }
jnxPTX5000SlotFPC      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 3 }
jnxPTX5000SlotFan      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 4 }
jnxPTX5000SlotCB       OBJECT IDENTIFIER ::= { jnxSlotPTX5000 5 }
jnxPTX5000SlotFPB      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 6 }
jnxPTX5000SlotSPMB     OBJECT IDENTIFIER ::= { jnxSlotPTX5000 7 }
jnxPTX5000SlotPDU      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 8 }
jnxPTX5000SlotPSM      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 9 }
jnxPTX5000SlotCCG      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 10 }
jnxPTX5000SlotPIC      OBJECT IDENTIFIER ::= { jnxSlotPTX5000 11 }

jnxMediaCardSpacePTX5000 OBJECT IDENTIFIER ::= { jnxMediaCardSpace 69 }
}
jnxPTX5000MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpacePTX5000
1 }

jnxMidplanePTX5000     OBJECT IDENTIFIER ::= { jnxBackplane      69 }

jnxModulePTX5000       OBJECT IDENTIFIER ::= { jnxModule        69 }
jnxPTX5000SIB          OBJECT IDENTIFIER ::= { jnxModulePTX5000 1 }
jnxPTX5000HM           OBJECT IDENTIFIER ::= { jnxModulePTX5000 2 }
jnxPTX5000FPC          OBJECT IDENTIFIER ::= { jnxModulePTX5000 3 }
jnxPTX5000Fan          OBJECT IDENTIFIER ::= { jnxModulePTX5000 4 }
jnxPTX5000CB           OBJECT IDENTIFIER ::= { jnxModulePTX5000 5 }

```

jnxPTX5000FPB	OBJECT IDENTIFIER ::= { jnxModulePTX5000 6 }
jnxPTX5000SPMB	OBJECT IDENTIFIER ::= { jnxModulePTX5000 7 }
jnxPTX5000PDU	OBJECT IDENTIFIER ::= { jnxModulePTX5000 8 }
jnxPTX5000PSM	OBJECT IDENTIFIER ::= { jnxModulePTX5000 9 }
jnxPTX5000CCG	OBJECT IDENTIFIER ::= { jnxModulePTX5000 10 }
jnxPTX5000PIC	OBJECT IDENTIFIER ::= { jnxModulePTX5000 11 }

MIB Objects for the QFX Series

This topic lists the Juniper Networks enterprise-specific SNMP Chassis MIB definition objects for the QFX Series:

- [QFX3500 and QFX3600 Switches on page 363](#)
- [QFabric Systems on page 363](#)
- [QFabric System QFX3100 Director Device on page 363](#)
- [QFabric System QFX3008-I Interconnect Device on page 363](#)
- [QFabric System QFX3600-I Interconnect Device on page 364](#)
- [QFabric System Node Devices on page 364](#)

QFX3500 and QFX3600 Switches

```

jnxProductLineQFXSwitch      OBJECT IDENTIFIER ::= { jnxProductLine      82 }
jnxProductNameQFXSwitch      OBJECT IDENTIFIER ::= { jnxProductName      82 }
jnxProductModelQFXSwitch     OBJECT IDENTIFIER ::= { jnxProductModel     82 }
jnxProductVariationQFXSwitch OBJECT IDENTIFIER ::= { jnxProductVariation 82 }
jnxProductQFX3500s           OBJECT IDENTIFIER ::= { jnxProductVariationQFXSwitch 1 }
jnxProductQFX360016QS        OBJECT IDENTIFIER ::= { jnxProductVariationQFXSwitch 2 }
jnxProductQFX350048T4QS       OBJECT IDENTIFIER ::= { jnxProductVariationQFXSwitch 3 }

jnxChassisQFXSwitch          OBJECT IDENTIFIER ::= { jnxChassis          82 }

jnxSlotQFXSwitch             OBJECT IDENTIFIER ::= { jnxSlot             82 }
jnxQFXSwitchSlotFPC          OBJECT IDENTIFIER ::= { jnxSlotQFXSwitch     1 }
jnxQFXSwitchSlotHM           OBJECT IDENTIFIER ::= { jnxSlotQFXSwitch     2 }
jnxQFXSwitchSlotPower        OBJECT IDENTIFIER ::= { jnxSlotQFXSwitch     3 }
jnxQFXSwitchSlotFan          OBJECT IDENTIFIER ::= { jnxSlotQFXSwitch     4 }
jnxQFXSwitchSlotFPB          OBJECT IDENTIFIER ::= { jnxSlotQFXSwitch     5 }

jnxMediaCardSpaceQFXSwitch   OBJECT IDENTIFIER ::= { jnxMediaCardSpace 82 }
jnxQFXSwitchMediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceQFXSwitch 1 }

```

QFabric Systems

```

jnxProductLineQFX3000        OBJECT IDENTIFIER ::= { jnxProductLine 84 }
jnxProductNameQFX3000        OBJECT IDENTIFIER ::= { jnxProductName 84 }
jnxProductModelQFX3000       OBJECT IDENTIFIER ::= { jnxProductModel 84 }
jnxProductVariationQFX3000   OBJECT IDENTIFIER ::= { jnxProductVariation 84 }
jnxProductQFX3000-G          OBJECT IDENTIFIER ::= { jnxProductVariationQFX3000 1 }
jnxProductQFX3000-M          OBJECT IDENTIFIER ::= { jnxProductVariationQFX3000 2 }
jnxChassisQFX3000            OBJECT IDENTIFIER ::= { jnxChassis      84 }

```

QFabric System QFX3100 Director Device

```

jnxProductLineQFX3100 OBJECT IDENTIFIER ::= { jnxProductLine      100 }
jnxProductNameQFX3100 OBJECT IDENTIFIER ::= { jnxProductName      100 }
jnxProductModelQFX3100 OBJECT IDENTIFIER ::= { jnxProductModel     100 }
jnxProductVariationQFX3100 OBJECT IDENTIFIER ::= { jnxProductVariation 100 }
jnxChassisQFX3100      OBJECT IDENTIFIER ::= { jnxChassis          100 }

jnxSlotQFX3100          OBJECT IDENTIFIER ::= { jnxSlot             100 }
jnxQFX3100SlotCPU       OBJECT IDENTIFIER ::= { jnxSlotQFX3100      1 }
jnxQFX3100SlotMemory    OBJECT IDENTIFIER ::= { jnxSlotQFX3100      2 }
jnxQFX3100SlotPower     OBJECT IDENTIFIER ::= { jnxSlotQFX3100      3 }
jnxQFX3100SlotFan       OBJECT IDENTIFIER ::= { jnxSlotQFX3100      4 }
jnxQFX3100SlotHardDisk  OBJECT IDENTIFIER ::= { jnxSlotQFX3100      5 }
jnxQFX3100SlotNIC       OBJECT IDENTIFIER ::= { jnxSlotQFX3100      6 }

```

QFabric System QFX3008-I Interconnect Device

```

jnxProductLineQFXInterconnect OBJECT IDENTIFIER ::= { jnxProductLine      60 }
jnxProductNameQFXInterconnect OBJECT IDENTIFIER ::= { jnxProductName      60 }
jnxProductModelQFXInterconnect OBJECT IDENTIFIER ::= { jnxProductModel     60 }
jnxProductVariationQFXInterconnect OBJECT IDENTIFIER ::= { jnxProductVariation 60 }
jnxProductQFX3008             OBJECT IDENTIFIER ::= { jnxProductVariationQFXInterconnect 1 }
jnxProductQFXC083008          OBJECT IDENTIFIER ::= { jnxProductVariationQFXInterconnect 2 }
jnxProductQFX3008I            OBJECT IDENTIFIER ::= { jnxProductVariationQFXInterconnect 3 }

```

```

jnxChassisQFXInterconnect    OBJECT IDENTIFIER ::= { jnxChassis          60 }

jnxSlotQFXInterconnect       OBJECT IDENTIFIER ::= { jnxSlot          60 }
  jnxQFXInterconnectSlotFPC   OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 1 }
  jnxQFXInterconnectSlotHM    OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 2 }
  jnxQFXInterconnectSlotPower OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 3 }
  jnxQFXInterconnectSlotFan   OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 4 }
  jnxQFXInterconnectSlotCBD   OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 5 }
  jnxQFXInterconnectSlotFPB   OBJECT IDENTIFIER ::= { jnxSlotQFXInterconnect 6 }

jnxMediaCardSpaceQFXInterconnect OBJECT IDENTIFIER ::= { jnxMediaCardSpace 60 }
  jnxQFXInterconnectMediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceQFXInterconnect 1 }

jnxMidplaneQFXInterconnect   OBJECT IDENTIFIER ::= { jnxBackplane        60 }

```

QFabric System QFX3600-I Interconnect Device

```

jnxProductLineQFXMInterconnect OBJECT IDENTIFIER ::= { jnxProductLine    91 }
jnxProductNameQFXMInterconnect OBJECT IDENTIFIER ::= { jnxProductName    91 }
jnxProductModelQFXMInterconnect OBJECT IDENTIFIER ::= { jnxProductModel  91 }
jnxProductVariationQFXMInterconnect OBJECT IDENTIFIER ::= { jnxProductVariation 91 }
  jnxProductQFX3600I          OBJECT IDENTIFIER ::= { jnxProductVariationQFXMInterconnect 1 }

jnxChassisQFXMInterconnect     OBJECT IDENTIFIER ::= { jnxChassis        91 }

jnxSlotQFXMInterconnect        OBJECT IDENTIFIER ::= { jnxSlot          91 }
  jnxQFXMInterconnectSlotFPC    OBJECT IDENTIFIER ::= { jnxSlotQFXMInterconnect 1 }
  jnxQFXMInterconnectSlotHM     OBJECT IDENTIFIER ::= { jnxSlotQFXMInterconnect 2 }
  jnxQFXMInterconnectSlotPower  OBJECT IDENTIFIER ::= { jnxSlotQFXMInterconnect 3 }
  jnxQFXMInterconnectSlotFan    OBJECT IDENTIFIER ::= { jnxSlotQFXMInterconnect 4 }
  jnxQFXMInterconnectSlotFPB    OBJECT IDENTIFIER ::= { jnxSlotQFXMInterconnect 5 }

jnxMediaCardSpaceQFXMInterconnect OBJECT IDENTIFIER ::= { jnxMediaCardSpace 91 }
  jnxQFXMInterconnectMediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceQFXMInterconnect 1 }

```

QFabric System Node Devices

```

jnxProductLineQFXNode         OBJECT IDENTIFIER ::= { jnxProductLine    61 }
jnxProductNameQFXNode         OBJECT IDENTIFIER ::= { jnxProductName    61 }
jnxProductModelQFXNode        OBJECT IDENTIFIER ::= { jnxProductModel  61 }
jnxProductVariationQFXNode    OBJECT IDENTIFIER ::= { jnxProductVariation 61 }
  jnxProductQFX3500           OBJECT IDENTIFIER ::= { jnxProductVariationQFXNode 1 }
  jnxProductQFX360016Q        OBJECT IDENTIFIER ::= { jnxProductVariationQFXNode 3 }

jnxChassisQFXNode             OBJECT IDENTIFIER ::= { jnxChassis        61 }

jnxSlotQFXNode                OBJECT IDENTIFIER ::= { jnxSlot          61 }
  jnxQFXNodeSlotFPC            OBJECT IDENTIFIER ::= { jnxSlotQFXNode    1 }
  jnxQFXNodeSlotHM             OBJECT IDENTIFIER ::= { jnxSlotQFXNode    2 }
  jnxQFXNodeSlotPower          OBJECT IDENTIFIER ::= { jnxSlotQFXNode    3 }
  jnxQFXNodeSlotFan            OBJECT IDENTIFIER ::= { jnxSlotQFXNode    4 }
  jnxQFXNodeSlotFPB            OBJECT IDENTIFIER ::= { jnxSlotQFXNode    5 }

jnxMediaCardSpaceQFXNode      OBJECT IDENTIFIER ::= { jnxMediaCardSpace 61 }
  jnxQFXNodeMediaCardSpacePIC  OBJECT IDENTIFIER ::= { jnxMediaCardSpaceQFXNode 1 }

```

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 }

```

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 }

```

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 }

```

MIB Objects for the SRX5600 Services Gateway

The Chassis MIB objects for the SRX5600 Services Gateway include:

```

jnxProductLineSRX5600    OBJECT IDENTIFIER ::= { jnxProductLine 28 }
jnxProductNameSRX5600    OBJECT IDENTIFIER ::= { jnxProductName 28 }
jnxProductModelSRX5600   OBJECT IDENTIFIER ::= { jnxProductModel 28 }
jnxProductVariationSRX5600 OBJECT IDENTIFIER ::= { jnxProductVariation 28 }
jnxChassisSRX5600        OBJECT IDENTIFIER ::= { jnxChassis 28 }

jnxSlotSRX5600           OBJECT IDENTIFIER ::= { jnxSlot 28 }
jnxSRX5600SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX5600 1 }
jnxSRX5600SlotHM         OBJECT IDENTIFIER ::= { jnxSlotSRX5600 2 }
jnxSRX5600SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX5600 3 }
jnxSRX5600SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX5600 4 }
jnxSRX5600SlotCB         OBJECT IDENTIFIER ::= { jnxSlotSRX5600 5 }
jnxSRX5600SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotSRX5600 6 }

jnxMediaCardSpaceSRX5600 OBJECT IDENTIFIER ::= { jnxMediaCardSpace 28
}
jnxSRX5600MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX5600 1
}

jnxMidplaneSRX5600      OBJECT IDENTIFIER ::= { jnxBackplane 28 }

```

MIB Objects for the SRX5800 Services Gateway

The Chassis MIB objects for the SRX5800 Services Gateway include:

```

jnxProductLineSRX5800    OBJECT IDENTIFIER ::= { jnxProductLine 26 }
jnxProductNameSRX5800    OBJECT IDENTIFIER ::= { jnxProductName 26 }
jnxProductModelSRX5800   OBJECT IDENTIFIER ::= { jnxProductModel 26 }
jnxProductVariationSRX5800 OBJECT IDENTIFIER ::= { jnxProductVariation 26 }
jnxChassisSRX5800        OBJECT IDENTIFIER ::= { jnxChassis 26 }

jnxSlotSRX5800           OBJECT IDENTIFIER ::= { jnxSlot 26 }
jnxSRX5800SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX5800 1 }
jnxSRX5800SlotHM         OBJECT IDENTIFIER ::= { jnxSlotSRX5800 2 }
jnxSRX5800SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX5800 3 }
jnxSRX5800SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX5800 4 }
jnxSRX5800SlotCB         OBJECT IDENTIFIER ::= { jnxSlotSRX5800 5 }
jnxSRX5800SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotSRX5800 6 }

```

```

jnxMediaCardSpaceSRX5800    OBJECT IDENTIFIER ::= { jnxMediaCardSpace    26
}
  jnxSRX5800MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX5800
1 }

jnxMidplaneSRX5800          OBJECT IDENTIFIER ::= { jnxBackplane          26 }

```

MIB Objects for the SRX100 Services Gateway

The Chassis MIB objects for the SRX100 Services Gateway include:

```

jnxProductLineSRX100    OBJECT IDENTIFIER ::= { jnxProductLine 41 }
jnxProductNameSRX100    OBJECT IDENTIFIER ::= { jnxProductName 41 }
jnxChassisSRX100        OBJECT IDENTIFIER ::= { jnxChassis      41 }

jnxSlotSRX100           OBJECT IDENTIFIER ::= { jnxSlot         41 }
jnxSRX100SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX100   1 }
jnxSRX100SlotRE         OBJECT IDENTIFIER ::= { jnxSlotSRX100   2 }
jnxSRX100SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX100   3 }
jnxSRX100SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX100   4 }

jnxMediaCardSpaceSRX100    OBJECT IDENTIFIER ::= { jnxMediaCardSpace    41
}
  jnxSRX100MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX100 1
}

jnxMidplaneSRX100          OBJECT IDENTIFIER ::= { jnxBackplane 41 }

jnxModuleSRX100           OBJECT IDENTIFIER ::= { jnxModule         41 }
jnxSRX100FPC              OBJECT IDENTIFIER ::= { jnxModuleSRX100 1 }
jnxSRX100RE               OBJECT IDENTIFIER ::= { jnxModuleSRX100 2 }
jnxSRX100Power            OBJECT IDENTIFIER ::= { jnxModuleSRX100 3 }
jnxSRX100Fan              OBJECT IDENTIFIER ::= { jnxModuleSRX100 4 }

```

MIB Objects for the SRX210 Services Gateway

The Chassis MIB objects for the SRX210 Services Gateway include:

```

jnxProductLineSRX210    OBJECT IDENTIFIER ::= { jnxProductLine    36 }
jnxProductNameSRX210    OBJECT IDENTIFIER ::= { jnxProductName    36 }
jnxChassisSRX210        OBJECT IDENTIFIER ::= { jnxChassis        36 }

jnxSlotSRX210           OBJECT IDENTIFIER ::= { jnxSlot           36 }
jnxSRX210SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotSRX210     1 }
jnxSRX210SlotRE         OBJECT IDENTIFIER ::= { jnxSlotSRX210     2 }
jnxSRX210SlotPower      OBJECT IDENTIFIER ::= { jnxSlotSRX210     3 }
jnxSRX210SlotFan        OBJECT IDENTIFIER ::= { jnxSlotSRX210     4 }

jnxMediaCardSpaceSRX210    OBJECT IDENTIFIER ::= { jnxMediaCardSpace    36
}
  jnxSRX210MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX210 1
}

jnxMidplaneSRX210          OBJECT IDENTIFIER ::= { jnxBackplane 36 }

jnxModuleSRX210           OBJECT IDENTIFIER ::= { jnxModule           36 }
jnxSRX210FPC              OBJECT IDENTIFIER ::= { jnxModuleSRX210 1 }
jnxSRX210RE               OBJECT IDENTIFIER ::= { jnxModuleSRX210 2 }

```

```
jnxSRX210Power      OBJECT IDENTIFIER ::= { jnxModuleSRX210 3 }
jnxSRX210Fan        OBJECT IDENTIFIER ::= { jnxModuleSRX210 4 }
```

MIB Objects for the SRX220 Services Gateway

The Chassis MIB objects for the SRX220 Services Gateway include:

```
jnxProductLineSRX220      OBJECT IDENTIFIER ::= { jnxProductLine      58 }
jnxProductNameSRX220      OBJECT IDENTIFIER ::= { jnxProductName      58 }
jnxChassisSRX220          OBJECT IDENTIFIER ::= { jnxChassis        58 }

jnxSlotSRX220             OBJECT IDENTIFIER ::= { jnxSlot          58 }
jnxSRX220SlotFPC          OBJECT IDENTIFIER ::= { jnxSlotSRX220    1 }
jnxSRX220SlotRE           OBJECT IDENTIFIER ::= { jnxSlotSRX220    2 }
jnxSRX220SlotPower        OBJECT IDENTIFIER ::= { jnxSlotSRX220    3 }
jnxSRX220SlotFan          OBJECT IDENTIFIER ::= { jnxSlotSRX220    4 }

jnxMediaCardSpaceSRX220   OBJECT IDENTIFIER ::= { jnxMediaCardSpace      58
}
jnxSRX220MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceSRX220
1 }

jnxMidplaneSRX220         OBJECT IDENTIFIER ::= { jnxBackplane 58 }

jnxModuleSRX220           OBJECT IDENTIFIER ::= { jnxModule      58 }
jnxSRX220FPC              OBJECT IDENTIFIER ::= { jnxModuleSRX220 1 }
jnxSRX220RE               OBJECT IDENTIFIER ::= { jnxModuleSRX220 2 }
jnxSRX220SlotPower        OBJECT IDENTIFIER ::= { jnxSlotSRX220   3 }
jnxSRX220SlotFan          OBJECT IDENTIFIER ::= { jnxSlotSRX220   4 }
```

MIB Objects for the SRX240 Services Gateway

The Chassis MIB objects for the SRX240 Services Gateway include:

```
jnxProductLineJSRX240     OBJECT IDENTIFIER ::= { jnxProductLine   37 }
jnxProductNameJSRX240     OBJECT IDENTIFIER ::= { jnxProductName   37 }
jnxChassisJSRX240         OBJECT IDENTIFIER ::= { jnxChassis     37 }
jnxSlotJSRX240            OBJECT IDENTIFIER ::= { jnxSlot        37 }
jnxJSRX240SlotFPC         OBJECT IDENTIFIER ::= { jnxSlotJSRX240  1 }
jnxJSRX240SlotRE          OBJECT IDENTIFIER ::= { jnxSlotJSRX240  2 }
jnxJSRX240SlotPower       OBJECT IDENTIFIER ::= { jnxSlotJSRX240  3 }
jnxJSRX240SlotFan         OBJECT IDENTIFIER ::= { jnxSlotJSRX240  4 }
jnxMediaCardSpaceJSRX240  OBJECT IDENTIFIER ::= { jnxMediaCardSpace   37 }
jnxJSRX240MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceJSRX240 1
}
jnxMidplaneJSRX240        OBJECT IDENTIFIER ::= { jnxBackplane   37 }
jnxModuleJSRX240          OBJECT IDENTIFIER ::= { jnxModule      37 }
jnxJSRX240FPC             OBJECT IDENTIFIER ::= { jnxModuleJSRX240 1 }
jnxJSRX240RE              OBJECT IDENTIFIER ::= { jnxModuleJSRX240 2 }
jnxJSRX240Power           OBJECT IDENTIFIER ::= { jnxModuleJSRX240 3 }
jnxJSRX240Fan             OBJECT IDENTIFIER ::= { jnxModuleJSRX240 4 }
```

MIB Objects for the SRX550 Services Gateway

The Chassis MIB objects for the SRX550 Services Gateway include:

```
jnxProductLineJSRX550     OBJECT IDENTIFIER ::= { jnxProductLine   86 }
jnxProductNameJSRX550     OBJECT IDENTIFIER ::= { jnxProductName   86 }
jnxChassisJSRX550         OBJECT IDENTIFIER ::= { jnxChassis     86 }
```


jnxSlotJSRX550	OBJECT IDENTIFIER ::= { jnxSlot 86 }
jnxJSRX550SlotFPC	OBJECT IDENTIFIER ::= { jnxSlotJSRX550 1 }
jnxJSRX550SlotRE	OBJECT IDENTIFIER ::= { jnxSlotJSRX550 2 }
jnxJSRX550SlotPower	OBJECT IDENTIFIER ::= { jnxSlotJSRX550 3 }
jnxJSRX550SlotFan	OBJECT IDENTIFIER ::= { jnxSlotJSRX550 4 }
jnxMediaCardSpaceJSRX550	OBJECT IDENTIFIER ::= { jnxMediaCardSpace 86 }
jnxJSRX550MediaCardSpacePIC 1}	OBJECT IDENTIFIER ::= { jnxMediaCardSpaceJSRX550 1 }
jnxMidplaneJSRX550	OBJECT IDENTIFIER ::= { jnxBackplane 86 }
jnxModuleJSRX550	OBJECT IDENTIFIER ::= { jnxModule 86 }
jnxJSRX550FPC	OBJECT IDENTIFIER ::= { jnxModuleJSRX550 1 }
jnxJSRX550RE	OBJECT IDENTIFIER ::= { jnxModuleJSRX550 2 }
jnxJSRX550Power	OBJECT IDENTIFIER ::= { jnxModuleJSRX550 3 }
jnxJSRX550Fan	OBJECT IDENTIFIER ::= { jnxModuleJSRX550 4 }

MIB Objects for the SRX650 Services Gateway

The Chassis MIB objects for the SRX650 Services Gateway include:

jnxProductLineJSRX650	OBJECT IDENTIFIER ::= { jnxProductLine 38 }
jnxProductNameJSRX650	OBJECT IDENTIFIER ::= { jnxProductName 38 }
jnxChassisJSRX650	OBJECT IDENTIFIER ::= { jnxChassis 38 }
jnxSlotJSRX650	OBJECT IDENTIFIER ::= { jnxSlot 38 }
jnxJSRX650SlotFPC	OBJECT IDENTIFIER ::= { jnxSlotJSRX650 1 }
jnxJSRX650SlotRE	OBJECT IDENTIFIER ::= { jnxSlotJSRX650 2 }
jnxJSRX650SlotPower	OBJECT IDENTIFIER ::= { jnxSlotJSRX650 3 }
jnxJSRX650SlotFan	OBJECT IDENTIFIER ::= { jnxSlotJSRX650 4 }
jnxMediaCardSpaceJSRX650	OBJECT IDENTIFIER ::= { jnxMediaCardSpace 38 }
jnxJSRX650MediaCardSpacePIC 1}	OBJECT IDENTIFIER ::= { jnxMediaCardSpaceJSRX650 1 }
jnxMidplaneJSRX650	OBJECT IDENTIFIER ::= { jnxBackplane 38 }
jnxModuleJSRX650	OBJECT IDENTIFIER ::= { jnxModule 38 }
jnxJSRX650FPC	OBJECT IDENTIFIER ::= { jnxModuleJSRX650 1 }
jnxJSRX650RE	OBJECT IDENTIFIER ::= { jnxModuleJSRX650 2 }
jnxJSRX650Power	OBJECT IDENTIFIER ::= { jnxModuleJSRX650 3 }
jnxJSRX650Fan	OBJECT IDENTIFIER ::= { jnxModuleJSRX650 4 }

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 16

Class-of-Service MIB

- [Interpreting the Enterprise-Specific Class-of-Service MIB on page 371](#)

Interpreting the Enterprise-Specific Class-of-Service MIB

- [Class-of-Service MIB on page 371](#)
- [jnxCosInvQstatTable on page 371](#)

Class-of-Service MIB

The Juniper Networks enterprise-specific Class-of-Service (CoS) MIB provides support for monitoring interface output queue statistics per interface and per forwarding class.

The CoS MIB is an object of the **jnxMibs** branch of the enterprise-specific MIB and has an object identifier of **{jnxMIB 15}**. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cos.txt.

For information about the enterprise-specific CoS MIB objects, see the following topic:

- [jnxCosInvQstatTable on page 371](#)

jnxCosInvQstatTable

This table provides output queue statistics for each interface. Only those physical interfaces that support output queue statistics appear in this table. This table provides the same information as the **jnxCosQstatTable**, but the information is ordered by queue number and interface index, instead of by interface index and queue number.

The entries in the **jnxCosInvQstatTable**, whose object identifier is **{jnxCos 6}**, are represented by **jnxCosInvQstatEntry**, whose object identifier is **{jnxCosIfqStatsTable 1}**, and are listed in [Table 95 on page 371](#).

Table 95: jnxCosInvQstatEntry

Object	Object Identifier	Description
jnxCosInvQstatQedPkts	jnxCosInvQstatEntry 1	The number of packets queued at the output queue.
jnxCosInvQstatQedPktRate	jnxCosInvQstatEntry 2	The rate (in pps) at which packets are queued at the output queue.

Table 95: jnxCosInvQstatEntry (continued)

Object	Object Identifier	Description
jnxCosInvQstatQedBytes	jnxCosInvQstatEntry 3	The number of bytes queued at the output queue.
jnxCosInvQstatQedByteRate	jnxCosInvQstatEntry 4	The rate (in bytes per second) at which bytes are queued at the output queue.
jnxCosInvQstatQedTxedPkts	jnxCosInvQstatEntry 5	The number of packets transmitted on the queue.
jnxCosInvQstatQedTxedPktRate	jnxCosInvQstatEntry 6	The packet transmission rate of the output queue (in pps).
jnxCosInvQstatQedTxedBytes	jnxCosInvQstatEntry 7	The number of bytes transmitted on the queue.
jnxCosInvQstatQedTxedByteRate	jnxCosInvQstatEntry 8	The byte transmission rate of the output queue (in bytes per second).
jnxCosInvQstatQedTailDropPkts	jnxCosInvQstatEntry 9	The number of packets tail dropped at the output queue.
jnxCosInvQstatQedTailDropPktRate	jnxCosInvQstatEntry 10	The tail drop packet rate (in pps) for the queue.
jnxCosInvQstatTotalRedDropPkts	jnxCosInvQstatEntry 11	The number of packets dropped on the interface due to random early detection (RED) at the output.
jnxCosInvQstatTotalRedDropPktRate	jnxCosInvQstatEntry 12	The most-recent estimate of the drop rate (in pps) for packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropPkts	jnxCosInvQstatEntry 13	The number of low PLP nonTCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropPktRate	jnxCosInvQstatEntry 14	The rate (in pps) at which low PLP nonTCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropPkts	jnxCosInvQstatEntry 15	The number of low PLP TCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropPktRate	jnxCosInvQstatEntry 16	The rate (in pps) at which low PLP TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropPkts	jnxCosInvQstatEntry 17	The number of high PLP nonTCP packets dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropPktRate	jnxCosInvQstatEntry 18	The rate (in pps) at which high PLP nonTCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropPkts	jnxCosInvQstatEntry 19	The number of high PLP TCP packets dropped on the interface due to RED at the output.

Table 95: jnxCosInvQstatEntry (*continued*)

Object	Object Identifier	Description
jnxCosInvQstatHpTcpRedDropPktRate	jnxCosInvQstatEntry 20	The rate (in pps) at which high PLP TCP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatTotalRedDropBytes	jnxCosInvQstatEntry 21	The number of bytes dropped on the interface due to RED at the output.
jnxCosInvQstatTotalRedDropByteRate	jnxCosInvQstatEntry 22	The rate (in bytes per second) at which bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropBytes	jnxCosInvQstatEntry 23	The number of low PLP nonTCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatLpNonTcpRedDropByteRate	jnxCosInvQstatEntry 24	The rate (in bytes per second) at which low PLP nonTCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropBytes	jnxCosInvQstatEntry 25	The number of low PLP TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatLpTcpRedDropByteRate	jnxCosInvQstatEntry 26	The rate (in bytes per second) at which low PLP TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropBytes	jnxCosInvQstatEntry 27	The number of high PLP nonTCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHpNonTcpRedDropByteRate	jnxCosInvQstatEntry 28	The rate (in bytes per second) at which high PLP nonTCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropBytes	jnxCosInvQstatEntry 29	The number of high PLP TCP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHpTcpRedDropByteRate	jnxCosInvQstatEntry 30	The rate (in bytes per second) at which high PLP TCP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatLpRedDropPkts	jnxCosInvQstatEntry 31	The number of low PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatLpRedDropPktRate	jnxCosInvQstatEntry 32	The rate (in pps) at which low PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropPkts	jnxCosInvQstatEntry 33	The number of medium-low PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatMLpRedDropPktRate	jnxCosInvQstatEntry 34	The rate (in pps) at which medium-low PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropPkts	jnxCosInvQstatEntry 35	The number of medium-high PLP packets dropped on the interface due to RED at the output.

Table 95: jnxCosInvQstatEntry (continued)

Object	Object Identifier	Description
jnxCosInvQstatMHPRedDropPktRate	jnxCosInvQstatEntry 36	The rate (in pps) at which medium-high PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatHPRedDropPkts	jnxCosInvQstatEntry 37	The number of high PLP packets dropped on the interface due to RED at the output.
jnxCosInvQstatHPRedDropPktRate	jnxCosInvQstatEntry 38	The rate (in pps) at which high PLP packets are dropped on the interface due to RED at the output.
jnxCosInvQstatLPRedDropBytes	jnxCosInvQstatEntry 39	The number of low PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatLPRedDropByteRate	jnxCosInvQstatEntry 40	The rate (in bytes per second) at which low PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatMLPRedDropBytes	jnxCosInvQstatEntry 41	The number of medium-low PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatMLPRedDropByteRate	jnxCosInvQstatEntry 42	The rate (in bytes per second) at which medium-low PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropBytes	jnxCosInvQstatEntry 43	The number of medium-high PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatMHPRedDropByteRate	jnxCosInvQstatEntry 44	The rate (in bytes per second) at which medium-high PLP bytes are dropped on the interface due to RED at the output.
jnxCosInvQstatHPRedDropBytes	jnxCosInvQstatEntry 45	The number of high PLP bytes dropped on the interface due to RED at the output.
jnxCosInvQstatHPRedDropByteRate	jnxCosInvQstatEntry 46	The rate (in bytes per second) at which high PLP bytes are dropped on the interface due to RED at the output.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Configuration Management MIB

- Interpreting the Enterprise-Specific Configuration Management MIB on page 375

Interpreting the Enterprise-Specific Configuration Management MIB

- Configuration Management MIB on page 375
- Configuration Management MIB Textual Conventions on page 375
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 376
- Rescue Configuration Change Management Objects on page 378
- Configuration Management Notifications on page 378

Configuration Management MIB

The Juniper Networks enterprise-specific Configuration Management MIB, whose object identifier is **{jnxMibs18}**, 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/junos13.2/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 375
- Configuration Change Management Objects and jnxCmCfgChgEventTable on page 376
- Rescue Configuration Change Management Objects on page 378
- Configuration Management Notifications on page 378

Configuration Management MIB Textual Conventions

Table 96 on page 376 contains information about the textual conventions used in the enterprise-specific Configuration Management MIB.

Table 96: Configuration Management MIB Textual Conventions

Object	Description	Syntax
jnxCmCfChgSource	Identifies the source of the configuration event.	This object uses the following integer values: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 1—Nonexistent 2—Updated

Configuration Change Management Objects and jnxCmCfChgEventTable

The configuration change management objects listed in [Table 97 on page 376](#) along with the objects in the **jnxCmCfChgEventTable** provide the configuration change history.

Each **jnxCmCfChg**, whose object identifier is **{jnxCfMgmt 1}**, contains the objects listed in [Table 97 on page 376](#).

Table 97: Configuration Change Management Objects

Object	Object ID	Description
jnxCmCfChgLatestIndex	jnxCmCfChg 1	Contains the index used in jnxCmCfChgEventTable to represent the latest configuration change event.
jnxCmCfChgLatestTime	jnxCmCfChg 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.
jnxCmCfChgLatestDate	jnxCmCfChg 3	Shows the date and time when the configuration was last modified.
jnxCmCfChgLatestSource	jnxCmCfChg 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 “Configuration Management MIB Textual Conventions” on page 375 .
jnxCmCfChgLatestUser	jnxCmCfChg 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.

Table 97: Configuration Change Management Objects (*continued*)

Object	Object ID	Description
<code>jnxCmCfgChgMaxEventEntries</code>	<code>jnxCmCfgChg 6</code>	Shows the maximum number of entries that <code>jnxCmCfgChgEventTable</code> can contain. Allowable range is 0 though 2147483647. When the number of entries in <code>jnxCmCfgChgEventTable</code> exceeds the maximum value set for <code>jnxCmCfgChgMaxEventEntries</code> , the latest entry displaces the oldest entry in the table.

For information about the `jnxCmCfgChgEventTable`, see the following section:

- [jnxCmCfgChgEventTable on page 377](#)

`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 97 on page 376](#) controls the number of entries stored in `jnxCmCfgChgEventTable`.

Each `jnxCmCfgChgEventEntry`, whose object identifier is `{jnxCmCfgChgEventTable 1}`, contains the objects listed in [Table 98 on page 377](#).

Table 98: `jnxCmCfgChgEventTable`

Object	Object ID	Description
<code>jnxCmCfgChgEventIndex</code>	<code>jnxCmCfgChgEventEntry 1</code>	Uniquely identifies a configuration change event. The SNMP process assigns monotonically increasing values to each event as it occurs. However, when the SNMP process is reset, the index values also are reset.
<code>jnxCmCfgChgEventTime</code>	<code>jnxCmCfgChgEventEntry 2</code>	Contains the value of <code>sysUpTime</code> when the event occurred.
<code>jnxCmCfgChgEventDate</code>	<code>jnxCmCfgChgEventEntry 3</code>	Contains the system date and time when the event occurred.
<code>jnxCmCfgChgEventSource</code>	<code>jnxCmCfgChgEventEntry 4</code>	Shows the source from which the configuration event was triggered. This object uses <code>JnxCmCfChgSource</code> to represent the source of configuration event. For more information about <code>JnxCmCfChgSource</code> , see “ Configuration Management MIB Textual Conventions ” on page 375 .
<code>jnxCmCfgChgEventUser</code>	<code>jnxCmCfgChgEventEntry 5</code>	Contains the name of the user who was logged in at the time of the event. Returns a zero-length string if the user name is not applicable or not available.
<code>jnxCmCfgChgEventLog</code>	<code>jnxCmCfgChgEventEntry 6</code>	Contains the log of the configuration event. Returns a zero-length string if no log is available.

Rescue Configuration Change Management Objects

The **jnxCmRescueChg**, whose object identifier is **{jnxCfgMgmt 2}**, contains information about changes to rescue configuration.

Table 99 on page 378 lists the objects associated with **jnxCmRescueChg**.

Table 99: 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 jnxCmCfgChgSource to represent the source of configuration event. For more information about jnxCmCfgChgSource , see “ Configuration Management MIB Textual Conventions ” on page 375.
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 “ Configuration Management MIB Textual Conventions ” on page 375.

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**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Destination Class Usage MIB

- [Interpreting the Enterprise-Specific Destination Class Usage MIB on page 381](#)

Interpreting the Enterprise-Specific Destination Class Usage MIB

- [Destination Class Usage MIB on page 381](#)
- [jnxDCUsTable on page 381](#)
- [jnxDcuStatsTable on page 382](#)

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 **{enterprise2636}** and has an object identifier of **{jnxMIB6}**. 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 *Routing Policy Feature Guide for Routing Devices* and *Junos OS Network Interfaces Library for Routing Devices*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/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 381](#)
- [jnxDcuStatsTable on page 382](#)

jnxDCUsTable

The entries in the **jnxDCUsTable**, whose object identifier is **{jnxDCUTable1}**, are represented by **jnxDCUsEntry** and are listed in [Table 100 on page 382](#).

Table 100: jnxDCUsEntry

Object	Object Identifier	Description
jnxDCUSrcIfIndex	jnxDCUsEntry 1	The interface index of the ingress interface.
jnxDCUDstClassName	jnxDCUsEntry 2	The destination class name specified in a routing policy and applied to the forwarding table.
jnxDCUPackets	jnxDCUsEntry 3	The number of packets passing through the network.
jnxDCUBytes	jnxDCUsEntry 4	The number of bytes passing through the network.

jnxDcuStatsTable

jnxDcuStatsTable contains statistics for traffic that satisfies the rules in each configured destination class. A separate set of statistics is kept for each destination class on each interface and address family on which this feature is enabled. This is essentially a replacement for **jnxDCUsTable**.

The entries in the **jnxDcuStatsTable**, whose object identifier is **{jnxDCUs 2}**, are represented by **jnxDCUsStatsEntry** and are listed in [Table 101 on page 382](#).

Table 101: jnxDCUsStatsEntry

Object	Object Identifier	Description
jnxDcuStatsSrcIfIndex	jnxDcuStatsEntry 1	The interface index of the ingress interface for traffic counted in each entry.
jnxDcuStatsAddrFamily	jnxDcuStatsEntry 2	The address family of the entry's traffic.
jnxDcuStatsClassName	jnxDcuStatsEntry 3	The name of the destination class that applies to the entry's traffic.
jnxDcuStatsPackets	jnxDcuStatsEntry 4	The number of packets received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsBytes	jnxDcuStatsEntry 5	The number of bytes received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsCIName	jnxDcuStatsEntry 6	The name of the destination class. This object is a duplicate of jnxDcuStatsClassName and is included to satisfy those network management applications that cannot extract the destination class name from the instance portion of the OID.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)

- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 19

DHCP MIB

- Interpreting the Enterprise-Specific DHCP MIB on page 385

Interpreting the Enterprise-Specific DHCP MIB

- DHCP MIB on page 385
- DHCP Local Server on page 385
- DHCP Relay on page 391

DHCP MIB

The Juniper Networks enterprise-specific DHCP MIB, whose object identifier is **{jnxMibs 61}**, stores both local server and relay objects. This MIB provides support for Junos OS DHCP for bindings and lease tables, and for statistics.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-jdhcp.txt.

The DHCP MIB contains two managed object groups:

- **jnxJdhcp** local server objects **{jnxJdhcpMib 1}**
- **jnxJdhcp** relay objects **{jnxJdhcpMib 2}**

For information about using the Traceroute MIB and enterprise-specific DHCP MIB, see the *Network Management Administration Guide for Routing Devices*.

For information about DHCP MIB objects, see the following topics:

- DHCP Local Server on page 385
- DHCP Relay on page 391

DHCP Local Server

The **jnxJdhcpLocalServerObjects** contains objects that describe the current statistics and bindings of a DHCP local server.

Object IDs for the DHCP local server are as follows:

- `jnxJdhcpLocalServerStatistics`—{`jnxJdhcpLocalServerObjects 1`}
- `jnxJdhcpLocalServerBindings`—{`jnxJdhcpLocalServerObjects 2`}
- `jnxJdhcpLocalServerTraps`—{`jnxJdhcpLocalServerObjects 3`}
- `jnxJdhcpLocalServerTrapVars`—{`jnxJdhcpLocalServerObjects 4`}
- `jnxJdhcpLocalServerIfcStats`—{`jnxJdhcpLocalServerObjects 5`}

The `jnxJdhcpLocalServerStatistics`, whose object ID is {`jnxJdhcpLocalServerObjects 1`}, contains objects that describe the current status and statistics of a DHCP local server.

The `jnxJdhcpLocalServerStatistics` objects are listed in [Table 102 on page 386](#).

Table 102: `jnxJdhcpLocalServerStatistics`

Object	Object Identifier	Description
<code>jnxJdhcpLocalServerTotalDropped</code>	<code>jnxJdhcpLocalServerStatistics 1</code>	Total number of packets dropped.
<code>jnxJdhcpLocalServerBadHardwareDropped</code>	<code>jnxJdhcpLocalServerStatistics 2</code>	Total number of packets dropped due to bad hardware address.
<code>jnxJdhcpLocalServerBadBootpOpcodeDropped</code>	<code>jnxJdhcpLocalServerStatistics 3</code>	Number of packets dropped due to no BOOTP message.
<code>jnxJdhcpLocalServerBadOptionsDropped</code>	<code>jnxJdhcpLocalServerStatistics 4</code>	Number of packets dropped due to bad options in the packet.
<code>jnxJdhcpLocalServerBadAddressDropped</code>	<code>jnxJdhcpLocalServerStatistics 5</code>	Number of packets dropped due to invalid server address.
<code>jnxJdhcpLocalServerNoAddressDropped</code>	<code>jnxJdhcpLocalServerStatistics 6</code>	Number of packets dropped due to no available addresses.
<code>jnxJdhcpLocalServerNoInterfaceDropped</code>	<code>jnxJdhcpLocalServerStatistics 7</code>	Number of packets dropped due to no interface match.
<code>jnxJdhcpLocalServerNoRoutingInstanceDropped</code>	<code>jnxJdhcpLocalServerStatistics 8</code>	Number of packets dropped due to no routing instance match.
<code>jnxJdhcpLocalServerNoLocalAddressDropped</code>	<code>jnxJdhcpLocalServerStatistics 9</code>	Number of packets dropped due to no valid local address.
<code>jnxJdhcpLocalServerShortPacketDropped</code>	<code>jnxJdhcpLocalServerStatistics 10</code>	Number of packets dropped due to packet being too short.
<code>jnxJdhcpLocalServerBadReadDropped</code>	<code>jnxJdhcpLocalServerStatistics 11</code>	Number of packets dropped due to read error.
<code>jnxJdhcpLocalServerBadSendDropped</code>	<code>jnxJdhcpLocalServerStatistics 12</code>	Number of packets dropped due to send error.

Table 102: jnxJdhcpLocalServerStatistics (*continued*)

Object	Object Identifier	Description
jnxJdhcpLocalServerAuthenticationDropped	jnxJdhcpLocalServerStatistics 13	Number of packets dropped due to authentication failure.
jnxJdhcpLocalServerDynamicProfileDropped	jnxJdhcpLocalServerStatistics 14	Number of packets dropped due to dynamic profile error.
jnxJdhcpLocalServerLicenseDropped	jnxJdhcpLocalServerStatistics 15	Number of packets dropped due to license error.
jnxJdhcpLocalServerBootRequestReceived	jnxJdhcpLocalServerStatistics 16	Number of Boot Request packets received.
jnxJdhcpLocalServerDhcpDeclineReceived	jnxJdhcpLocalServerStatistics 17	Number of DHCP Decline packets received.
jnxJdhcpLocalServerDhcpDiscoverReceived	jnxJdhcpLocalServerStatistics 18	Number of DHCP Discover packets received.
jnxJdhcpLocalServerDhcpInformReceived	jnxJdhcpLocalServerStatistics 19	Number of DHCP inform packets received.
jnxJdhcpLocalServerDhcpReleaseReceived	jnxJdhcpLocalServerStatistics 20	Number of DHCP release packets received.
jnxJdhcpLocalServerDhcpRequestReceived	jnxJdhcpLocalServerStatistics 21	Number of DHCP request packets received.
jnxJdhcpLocalServerDhcpBootReplySent	jnxJdhcpLocalServerStatistics 22	Number of DHCP Boot Reply packets sent.
jnxJdhcpLocalServerDhcpOfferSent	jnxJdhcpLocalServerStatistics 23	Number of DHCP Offer packets sent.
jnxJdhcpLocalServerDhcpAckSent	jnxJdhcpLocalServerStatistics 24	Number of DHCP Ack packets sent.
jnxJdhcpLocalServerDhcpNakSent	jnxJdhcpLocalServerStatistics 25	Number of DHCP Nak packets sent.
jnxJdhcpLocalServerForceRenewSent	jnxJdhcpLocalServerStatistics 26	Number of DHCP Force Renew packets sent.

The `jnxJdhcpLocalServerBindings`, whose object ID is `{jnxJdhcpLocalServerObjects 2}`, contains information about bindings managed by a DHCP local server.

The `jnxJdhcpLocalServerBindings` objects are listed in [Table 103 on page 388](#).

Table 103: jnxJdhcpLocalServerBindings

Object	Object Identifier	Description
jnxJdhcpLocalServerBindingsIpAddress	jnxJdhcpLocalServerBindingsEntry 1	IP address associated with this entry in the bindings table.
jnxJdhcpLocalServerBindingsMacAddress	jnxJdhcpLocalServerBindingsEntry 2	MAC address associated with this entry in the bindings table and corresponding to the IP address denoted by the table index.
jnxJdhcpLocalServerBindingsState	jnxJdhcpLocalServerBindingsEntry 3	State associated with this entry in the bindings table.
jnxJdhcpLocalServerBindingsLeaseEndTime	jnxJdhcpLocalServerBindingsEntry 4	Time the lease expires on this binding.
jnxJdhcpLocalServerBindingsLeaseExpireTime	jnxJdhcpLocalServerBindingsEntry 5	Ttime remaining until the lease expires for this binding.
jnxJdhcpLocalServerBindingsLeaseStartTime	jnxJdhcpLocalServerBindingsEntry 6	Ttime the lease was started for this binding.
jnxJdhcpLocalServerBindingsIncomingClientInterface	jnxJdhcpLocalServerBindingsEntry 7	Incoming interface for this binding.
jnxJdhcpLocalServerBindingsClientInterfaceVlanId	jnxJdhcpLocalServerBindingsEntry 8	VLAN ID for this binding.
jnxJdhcpLocalServerBindingsDemuxInterfaceName	jnxJdhcpLocalServerBindingsEntry 9	Demux interface for this binding.
jnxJdhcpLocalServerBindingsServerIpAddress	jnxJdhcpLocalServerBindingsEntry 10	IP address associated with the server for this entry in the bindings table.
jnxJdhcpLocalServerBindingsBootpRelayAddress	jnxJdhcpLocalServerBindingsEntry 11	BOOTP relay address associated with the server for this entry in the bindings table.
jnxJdhcpLocalServerBindingsPreviousBootpRelayAddress	jnxJdhcpLocalServerBindingsEntry 12	Previous BOOTP relay address associated with the server for this entry in the bindings table.
jnxJdhcpLocalServerBindingsClientPoolName	jnxJdhcpLocalServerBindingsEntry 13	Client pool name for this binding.
jnxJdhcpLocalServerBindingsClientProfileName	jnxJdhcpLocalServerBindingsEntry 14	Client profile name for this binding.

The `jnxJdhcpLocalServerTraps`, whose object ID is `{jnxJdhcpLocalServerObjects 3}` contains objects that describe the DHCP local server notifications.

The `jnxJdhcpLocalServerTraps` objects are listed in [“Juniper Networks Enterprise-Specific SNMP Traps” on page 57](#).

`jnxJdhcpLocalServerTrapVars`, whose object ID is `{jnxJdhcpLocalServerObjects 4}`, contains objects that describe the contents of a table of interface statistics maintained by the DHCP local server.

`jnxJdhcpLocalServerTrapVars` objects are listed in [Table 104 on page 389](#).

Table 104: jnxJdhcpLocalServerTrapVars

Object	Object Identifier	Description
<code>jnxJdhcpLocalServerLastDetected</code>	<code>jnxJdhcpLocalServerTrapVars 1</code>	Time the duplicate client was last detected.
<code>jnxJdhcpLocalServerRouterName</code>	<code>jnxJdhcpLocalServerTrapVars 2</code>	VRF ID in Junos OS. Represented as the logical router (LR) name followed by the router instance (RI) name.
<code>jnxJdhcpLocalServerMacAddress</code>	<code>jnxJdhcpLocalServerTrapVars 3</code>	MAC address of the client that changed interfaces."
<code>jnxJdhcpLocalServerInterfaceName</code>	<code>jnxJdhcpLocalServerTrapVars 4</code>	Interface where the DHCP client was detected.
<code>jnxJdhcpLocalServerInterfaceLimit</code>	<code>jnxJdhcpLocalServerTrapVars 5</code>	Number of clients supported on this interface."

`jnxJdhcpLocalServerIfcStats`, whose object ID is `{jnxJdhcpLocalServerObjects 5}`, contains objects that describe the contents of a table of interface statistics maintained by the DHCP local server.

`jnxJdhcpLocalServerIfcStats` objects are listed in [Table 105 on page 389](#).

Table 105: jnxJdhcpLocalServerIfcStats

Object	Object Identifier	Description
<code>jnxJdhcpLocalServerIfcStatsIfIndex</code>	<code>jnxJdhcpLocalServerIfcStatsEntry 1</code>	IfIndex value of the interface for which this entry contains information.
<code>jnxJdhcpLocalServerIfcStatsTotalDropped</code>	<code>jnxJdhcpLocalServerIfcStatsEntry 2</code>	Total number of packets dropped.
<code>jnxJdhcpLocalServerIfcStatsBadHardwareDropped</code>	<code>jnxJdhcpLocalServerIfcStatsEntry 3</code>	Number of packets dropped due to bad hardware address.
<code>jnxJdhcpLocalServerIfcStatsBadBootpOpcodeDropped</code>	<code>jnxJdhcpLocalServerIfcStatsEntry 4</code>	Number of packets dropped due to no BOOTP message.
<code>jnxJdhcpLocalServerIfcStatsBadOptionsDropped</code>	<code>jnxJdhcpLocalServerIfcStatsEntry 5</code>	Number of packets dropped due to bad options in the packet.

Table 105: jnxJdhcpLocalServerIfcStats (*continued*)

Object	Object Identifier	Description
jnxJdhcpLocalServerIfcStatsBadAddressDropped	jnxJdhcpLocalServerIfcStatsEntry 6	Number of packets dropped due to invalid server address.
jnxJdhcpLocalServerIfcStatsNoAddressDropped	jnxJdhcpLocalServerIfcStatsEntry 7	Number of packets dropped due to no available addresses.
jnxJdhcpLocalServerIfcStatsNoInterfaceCfgDropped	jnxJdhcpLocalServerIfcStatsEntry 8	Number of packets dropped due to no interface match.
jnxJdhcpLocalServerIfcStatsNoLocalAddressDropped	jnxJdhcpLocalServerIfcStatsEntry 9	Number of packets dropped due to no valid local address.
jnxJdhcpLocalServerIfcStatsShortPacketDropped	jnxJdhcpLocalServerIfcStatsEntry 10	Number of packets dropped due to packet too short.
jnxJdhcpLocalServerIfcStatsBadSendDropped	jnxJdhcpLocalServerIfcStatsEntry 11	Number of packets dropped due to send error.
jnxJdhcpLocalServerIfcStatsAuthenticationDropped	jnxJdhcpLocalServerIfcStatsEntry 12	Number of packets dropped due to authentication failure.
jnxJdhcpLocalServerIfcStatsDynamicProfileDropped	jnxJdhcpLocalServerIfcStatsEntry 13	Number of packets dropped due to dynamic profile error.
jnxJdhcpLocalServerIfcStatsLicenseDropped	jnxJdhcpLocalServerIfcStatsEntry 14	Number of packets dropped due to license error.
jnxJdhcpLocalServerIfcStatsBootRequestReceived	jnxJdhcpLocalServerIfcStatsEntry 15	Number of Boot Request packets received.
jnxJdhcpLocalServerIfcStatsDhcpDeclineReceived	jnxJdhcpLocalServerIfcStatsEntry 16	Number of DHCP Decline packets received.
jnxJdhcpLocalServerIfcStatsDhcpDiscoverReceived	jnxJdhcpLocalServerIfcStatsEntry 17	Number of DHCP Discover packets received.
jnxJdhcpLocalServerIfcStatsDhcpInformReceived	jnxJdhcpLocalServerIfcStatsEntry 18	Number of DHCP Inform packets received.
jnxJdhcpLocalServerIfcStatsDhcpReleaseReceived	jnxJdhcpLocalServerIfcStatsEntry 19	Number of DHCP Release packets received.
jnxJdhcpLocalServerIfcStatsDhcpRequestReceived	jnxJdhcpLocalServerIfcStatsEntry 20	Number of DHCP Request packets received.
jnxJdhcpLocalServerIfcStatsDhcpBootReplySent	jnxJdhcpLocalServerIfcStatsEntry 21	Number of DHCP Boot Reply packets sent.

Table 105: jnxJdhcpLocalServerIfcStats (*continued*)

Object	Object Identifier	Description
jnxJdhcpLocalServerIfcStatsDhcpOfferSent	jnxJdhcpLocalServerIfcStatsEntry 22	Number of DHCP Offer packets sent.
jnxJdhcpLocalServerIfcStatsDhcpAckSent	jnxJdhcpLocalServerIfcStatsEntry 23	Number of DHCP Ack packets sent.
jnxJdhcpLocalServerIfcStatsDhcpNakSent	jnxJdhcpLocalServerIfcStatsEntry 24	Number of DHCP Nak packets sent.
jnxJdhcpLocalServerIfcStatsForceRenewSent	jnxJdhcpLocalServerIfcStatsEntry 25	Number of DHCP Force Renew packets sent.
jnxJdhcpLocalServerIfcStatsTotalLeaseCount	jnxJdhcpLocalServerIfcStatsEntry 26	Number of Bound DHCP clients.
jnxJdhcpLocalServerIfcStatsBadDhcpOpcodeDropped	jnxJdhcpLocalServerIfcStatsEntry 27	Number of packets dropped with bad DHCP opcode.
jnxJdhcpLocalServerIfcStatsNoOptionsDropped	jnxJdhcpLocalServerIfcStatsEntry 28	Number of packets dropped with no options.
jnxJdhcpLocalServerIfcStatsHopLimitDropped	jnxJdhcpLocalServerIfcStatsEntry 29	Number of packets dropped due to hop limit violation.
jnxJdhcpLocalServerIfcStatsTtlExpiredDropped	jnxJdhcpLocalServerIfcStatsEntry 30	Number of packets dropped due to TTL expiration.
jnxJdhcpLocalServerIfcStatsBadUdpChecksumDropped	jnxJdhcpLocalServerIfcStatsEntry 31	Number of packets dropped due to bad UDP checksum.
jnxJdhcpLocalServerIfcStatsOption60Dropped	jnxJdhcpLocalServerIfcStatsEntry 32	Number of packets dropped due to bad option 60.

DHCP Relay

The **jnxJdhcpRelayServerObjects** contains objects that describe the current statistics and bindings of a DHCP relay server.

Object IDs for DHCP relay server are as follows:

- **jnxJdhcpRelayStatistics**—{**jnxJdhcpRelayObjects 1**}
- **jnxJdhcpRelayBindings**—{**jnxJdhcpRelayObjects 2**}
- **jnxJdhcpRelayIfcStats**—{**jnxJdhcpRelayObjects 3**}

The **jnxJdhcpRelayStatistics**, whose object ID is {**jnxJdhcpRelayObjects 1**}, contains objects that describe the current status and statistics of a DHCP relay.

The **jnxJdhcpRelayStatistics** objects are listed in [Table 106 on page 392](#).

Table 106: jnxJdhcpRelayStatistics

Object	Object Identifier	Description
jnxJdhcpRelayTotalDropped	jnxJdhcpRelayStatistics 1	Total number of packets dropped.
jnxJdhcpRelayBadHardwareDropped	jnxJdhcpRelayStatistics 2	Total number of packets dropped due to bad hardware address.
jnxJdhcpRelayBadBootpOpcodeDropped	jnxJdhcpRelayStatistics 3	Number of packets dropped due to no BOOTP message.
jnxJdhcpRelayBadOptionsDropped	jnxJdhcpRelayStatistics 4	Number of packets dropped due to bad options in the packet.
jnxJdhcpRelayBadAddressDropped	jnxJdhcpRelayStatistics 5	Number of packets dropped due to invalid server address.
jnxJdhcpRelayNoAddressDropped	jnxJdhcpRelayStatistics 6	Number of packets dropped due to no available addresses.
jnxJdhcpRelayNoInterfaceDropped	jnxJdhcpRelayStatistics 7	Number of packets dropped due to no interface match.
jnxJdhcpRelayNoRoutingInstanceDropped	jnxJdhcpRelayStatistics 8	Number of packets dropped due to no routing instance match.
jnxJdhcpRelayNoLocalAddressDropped	jnxJdhcpRelayStatistics 9	Number of packets dropped due to no valid local address.
jnxJdhcpRelayShortPacketDropped	jnxJdhcpRelayStatistics 10	Number of packets dropped due to packet being too short.
jnxJdhcpRelayBadReadDropped	jnxJdhcpRelayStatistics 11	Number of packets dropped due to read error.
jnxJdhcpRelayBadSendDropped	jnxJdhcpRelayStatistics 12	Number of packets dropped due to send error.
jnxJdhcpRelayOption82Dropped	jnxJdhcpRelayStatistics 13	Number of packets dropped due to failure to add Option 82.
jnxJdhcpRelayOption60Dropped	jnxJdhcpRelayStatistics 14	Number of packets dropped due to failure to add Option 60.
jnxJdhcpRelayAuthenticationDropped	jnxJdhcpRelayStatistics 15	Number of packets dropped due to authentication failure.
jnxJdhcpRelayDynamicProfileDropped	jnxJdhcpRelayStatistics 16	Number of packets dropped due to dynamic profile error.
jnxJdhcpRelayLicenseDropped	jnxJdhcpRelayStatistics 17	Number of packets dropped due to license error.
jnxJdhcpRelayBootRequestReceived	jnxJdhcpRelayStatistics 18	Number of Boot Request packets received.

Table 106: jnxJdhcpRelayStatistics (*continued*)

Object	Object Identifier	Description
jnxJdhcpRelayDhcpDeclineReceived	jnxJdhcpRelayStatistics 19	Number of DHCP Decline packets received.
jnxJdhcpRelayDhcpDiscoverReceived	jnxJdhcpRelayStatistics 20	Number of DHCP Discover packets received.
jnxJdhcpRelayDhcpInformReceived	jnxJdhcpRelayStatistics 21	Number of DHCP Inform packets received.
jnxJdhcpRelayDhcpReleaseReceived	jnxJdhcpRelayStatistics 22	Number of DHCP Release packets received.
jnxJdhcpRelayDhcpRequestReceived	jnxJdhcpRelayStatistics 23	Number of DHCP Request packets received.
jnxJdhcpRelayDhcpBootReplySent	jnxJdhcpRelayStatistics 24	Number of DHCP Boot Reply packets sent.
jnxJdhcpRelayDhcpOfferSent	jnxJdhcpRelayStatistics 25	Number of DHCP Offer packets sent.
jnxJdhcpRelayDhcpAckSent	jnxJdhcpRelayStatistics 26	Number of DHCP Ack packets sent.
jnxJdhcpRelayDhcpNakSent	jnxJdhcpRelayStatistics 27	Number of DHCP Nak packets sent.
jnxJdhcpRelayForceRenewSent	jnxJdhcpRelayStatistics 28	Number of DHCP Force Renew packets sent.

The **jnxJdhcpRelayBindings**, whose object ID is **{jnxJdhcpRelayObjects 2}**, contains objects that describe the current statistics and bindings of a DHCP relay server.

The **jnxJdhcpRelayBindings** objects are listed in [Table 107 on page 393](#).

Table 107: jnxJdhcpRelayBindings

Object	Object Identifier	Description
jnxJdhcpRelayBindingsIpAddress	jnxJdhcpRelayBindingsEntry 1	IP address associated with this entry in the bindings table.
jnxJdhcpRelayBindingsLeaseState	jnxJdhcpRelayBindingsEntry 2	State associated with this entry in the bindings table.
jnxJdhcpRelayBindingsLeaseEndTime	jnxJdhcpRelayBindingsEntry 3	Time the lease expires on this binding.
jnxJdhcpRelayBindingsLeaseExpireTime	jnxJdhcpRelayBindingsEntry 4	Time remaining until the lease expires for this binding.
jnxJdhcpRelayBindingsLeaseStartTime	jnxJdhcpRelayBindingsEntry 5	Time the lease started for this binding.
jnxJdhcpRelayBindingsIncomingClientInterface	jnxJdhcpRelayBindingsEntry 6	Incoming interface for this binding.
jnxJdhcpRelayBindingsClientInterfaceVlanId	jnxJdhcpRelayBindingsEntry 7	VLAN ID for this binding.
jnxJdhcpRelayBindingsDemuxInterfaceName	jnxJdhcpRelayBindingsEntry 8	Demux interface for this binding.

Table 107: jnxJdhcpRelayBindings (*continued*)

Object	Object Identifier	Description
jnxJdhcpRelayBindingsServerIpAddress	jnxJdhcpRelayBindingsEntry 9	IP address associated with the server for this entry in the bindings table.
jnxJdhcpRelayBindingsServerInterface	jnxJdhcpRelayBindingsEntry 10	Demux server for this binding.
jnxJdhcpRelayBindingsBootpRelayAddress	jnxJdhcpRelayBindingsEntry 11	IP address associated with the BOOTP relay for this entry in the bindings table.
jnxJdhcpRelayBindingsPreviousBootpRelayAddress	jnxJdhcpRelayBindingsEntry 12	IP address associated with the previous BOOTP relay for this entry in the bindings table.
jnxJdhcpRelayBindingsClientProfileName	jnxJdhcpRelayBindingsEntry 13	Client profile name.

jnxJdhcpRelayIfcStats, whose object ID is {jnxJdhcpRelayObjects 3}, contains objects that describe the contents of a table of interface statistics maintained by the DHCP relay.

jnxJdhcpRelayIfcStats objects are listed in [Table 108 on page 394](#).

Table 108: jnxJdhcpRelayIfcStats

Object	Object Identifier	Description
jnxJdhcpRelayIfcStatsIfIndex	jnxJdhcpRelayIfcStatsEntry 1	IfIndex value of the interface for which this entry contains information.
jnxJdhcpRelayIfcStatsTotalDropped	jnxJdhcpRelayIfcStatsEntry 2	Total number of packets dropped.
jnxJdhcpRelayIfcStatsBadHardwareDropped	jnxJdhcpRelayIfcStatsEntry 3	Total number of packets dropped due to bad hardware address.
jnxJdhcpRelayIfcStatsBadBootpOpcodeDropped	jnxJdhcpRelayIfcStatsEntry 4	Number of packets dropped due to no BOOTP message.
jnxJdhcpRelayIfcStatsBadOptionsDropped	jnxJdhcpRelayIfcStatsEntry 5	Number of packets dropped due to bad options in the packet.
jnxJdhcpRelayIfcStatsBadAddressDropped	jnxJdhcpRelayIfcStatsEntry 6	Number of packets dropped due to invalid server address.
jnxJdhcpRelayIfcStatsNoAddressDropped	jnxJdhcpRelayIfcStatsEntry 7	Number of packets dropped due to no available addresses.
jnxJdhcpRelayIfcStatsNoInterfaceCfgDropped	jnxJdhcpRelayIfcStatsEntry 8	Number of packets dropped due to no interface match.
jnxJdhcpRelayIfcStatsNoLocalAddressDropped	jnxJdhcpRelayIfcStatsEntry 9	The number of packets dropped due to no valid local address.

Table 108: jnxJdhcpRelayIfcStats (continued)

Object	Object Identifier	Description
jnxJdhcpRelayIfcStatsShortPacketDropped	jnxJdhcpRelayIfcStatsEntry 10	Number of packets dropped due to packet too short.
jnxJdhcpRelayIfcStatsBadSendDropped	jnxJdhcpRelayIfcStatsEntry 11	Number of packets dropped due to send error.
jnxJdhcpRelayIfcStatsAuthenticationDropped	jnxJdhcpRelayIfcStatsEntry 12	Number of packets dropped due to authentication failure.
jnxJdhcpRelayIfcStatsDynamicProfileDropped	jnxJdhcpRelayIfcStatsEntry 13	Number of packets dropped due to dynamic profile error.
jnxJdhcpRelayIfcStatsLicenseDropped	jnxJdhcpRelayIfcStatsEntry 14	Number of packets dropped due to license error.
jnxJdhcpRelayIfcStatsBootRequestReceived	jnxJdhcpRelayIfcStatsEntry 15	Number of Boot Request packets received.
jnxJdhcpRelayIfcStatsDhcpDeclineReceived	jnxJdhcpRelayIfcStatsEntry 16	Number of DHCP Decline packets received.
jnxJdhcpRelayIfcStatsDhcpDiscoverReceived	jnxJdhcpRelayIfcStatsEntry 17	Number of DHCP Discover packets received.
jnxJdhcpRelayIfcStatsDhcpInformReceived	jnxJdhcpRelayIfcStatsEntry 18	Number of DHCP Inform packets received.
jnxJdhcpRelayIfcStatsDhcpReleaseReceived	jnxJdhcpRelayIfcStatsEntry 19	Number of DHCP Release packets received.
jnxJdhcpRelayIfcStatsDhcpRequestReceived	jnxJdhcpRelayIfcStatsEntry 20	Number of DHCP Request packets received.
jnxJdhcpRelayIfcStatsDhcpBootReplySent	jnxJdhcpRelayIfcStatsEntry 21	Number of DHCP Boot Reply packets sent.
jnxJdhcpRelayIfcStatsDhcpOfferSent	jnxJdhcpRelayIfcStatsEntry 22	Number of DHCP Offer packets sent.
jnxJdhcpRelayIfcStatsDhcpAckSent	jnxJdhcpRelayIfcStatsEntry 23	Number of DHCP Ack packets sent.
jnxJdhcpRelayIfcStatsDhcpNakSent	jnxJdhcpRelayIfcStatsEntry 24	Number of DHCP Nak packets sent.
jnxJdhcpRelayIfcStatsForceRenewSent	jnxJdhcpRelayIfcStatsEntry 25	Number of DHCP Force Renew packets sent.
jnxJdhcpRelayIfcStatsTotalLeaseCount	jnxJdhcpRelayIfcStatsEntry 26	Number of Bound DHCP clients.
jnxJdhcpRelayIfcStatsBadDhcpOpcodeDropped	jnxJdhcpRelayIfcStatsEntry 27	Number of packets dropped with bad DHCP opcode.
jnxJdhcpRelayIfcStatsNoOptionsDropped	jnxJdhcpRelayIfcStatsEntry 28	Number of packets dropped with no options.

Table 108: jnxJdhcpRelayIfcStats (continued)

Object	Object Identifier	Description
jnxJdhcpRelayIfcStatsHopLimitDropped	jnxJdhcpRelayIfcStatsEntry 29	Number of packets dropped due to hop limit violation.
jnxJdhcpRelayIfcStatsTtlExpiredDropped	jnxJdhcpRelayIfcStatsEntry 30	Number of packets dropped due to TTL expiration.
jnxJdhcpRelayIfcStatsBadUdpCksumDropped	jnxJdhcpRelayIfcStatsEntry 31	Number of packets dropped due to bad UDP checksum.
jnxJdhcpRelayIfcStatsOption82Dropped	jnxJdhcpRelayIfcStatsEntry 32	Number of packets dropped due to failure to add option 82.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 20

DHCPv6 MIB

- [Interpreting the Enterprise-Specific DHCPv6 MIB on page 397](#)

Interpreting the Enterprise-Specific DHCPv6 MIB

- [DHCPv6 MIB on page 397](#)
- [DHCPv6 Local Server on page 397](#)

DHCPv6 MIB

The Juniper Networks enterprise-specific DHCPv6 MIB, whose object identifier is **{jnxMibs 62}**, stores local server objects. This MIB provides support for Junos OS DHCPv6 for bindings and lease tables, and for statistics.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-jdhcpv6.txt.

The DHCPv6 MIB contains the following object group:

- **jnxJdhcpv6** local server objects **{jnxJdhcpv6Mib 1}**

For information about DHCPv6 MIB objects, see the following topic:

- [DHCPv6 Local Server on page 397](#)

DHCPv6 Local Server

The **jnxJdhcpv6LocalServerObjects** contains objects that describe the current statistics and bindings of a DHCPv6 local server.

Object IDs for the DHCPv6 local server are as follows:

- **jnxJdhcpv6LocalServerStatistics**—**{jnxJdhcpv6LocalServerObjects 1}**
- **jnxJdhcpv6LocalServerBindings**—**{jnxJdhcpv6LocalServerObjects 2}**
- **jnxJdhcpv6LocalServerTraps**—**{jnxJdhcpv6LocalServerObjects 3}**
- **jnxJdhcpv6LocalServerTrapVars**—**{jnxJdhcpv6LocalServerObjects 4}**
- **jnxJdhcpv6LocalServerIfcStats**—**{jnxJdhcpv6LocalServerObjects 5}**

The `jnxJdhcpv6LocalServerStatistics`, whose object ID is `{jnxJdhcpv6LocalServerObjects 1}`, contains objects that describe the current status and statistics of a DHCPv6 local server.

The `jnxJdhcpv6LocalServerStatistics` objects are listed in [Table 109 on page 398](#).

Table 109: jnxJdhcpv6LocalServerStatistics

Object	Object Identifier	Description
<code>jnxJdhcpv6LocalServerTotalDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 1</code>	Number of DHCPv6 packets dropped.
<code>jnxJdhcpv6LocalServerNoSafdDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 2</code>	Number of DHCPv6 packets dropped due to no safd match.
<code>jnxJdhcpv6LocalServerBadSendDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 3</code>	Number of DHCPv6 packets dropped due to send error.
<code>jnxJdhcpv6LocalServerShortPacketDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 4</code>	Number of DHCPv6 packets dropped due to packet being too short.
<code>jnxJdhcpv6LocalServerBadMsgtypeDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 5</code>	Number of DHCPv6 packets dropped due to bad opcode in the packet.
<code>jnxJdhcpv6LocalServerBadOptionsDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 6</code>	Number of DHCPv6 packets dropped due to bad options in the packet.
<code>jnxJdhcpv6LocalServerBadSrcAddressDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 7</code>	Number of DHCPv6 packets dropped due to invalid addr family.
<code>jnxJdhcpv6LocalServerRelayHopCountDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 8</code>	Number of DHCPv6 packets dropped due to max relays supported.
<code>jnxJdhcpv6LocalServerNoClientIdDropped</code>	<code>jnxJdhcpv6LocalServerStatistics 9</code>	Number of DHCPv6 packets dropped due to missing client ID.
<code>jnxJdhcpv6LocalServerDeclineReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 10</code>	Number of DHCPv6 Decline packets received.
<code>jnxJdhcpv6LocalServerSolicitReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 11</code>	Number of DHCPv6 Solicit packets received.
<code>jnxJdhcpv6LocalServerInformationRequestReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 12</code>	The number of DHCPv6 Information Request packets received.
<code>jnxJdhcpv6LocalServerReleaseReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 13</code>	Number of DHCPv6 Release packets received.
<code>jnxJdhcpv6LocalServerRequestReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 14</code>	Number of DHCPv6 Request packets received.
<code>jnxJdhcpv6LocalServerConfirmReceived</code>	<code>jnxJdhcpv6LocalServerStatistics 15</code>	Number of DHCPv6 Confirm packets received.

Table 109: jnxJdhcpv6LocalServerStatistics (*continued*)

Object	Object Identifier	Description
jnxJdhcpv6LocalServerRenewReceived	jnxJdhcpv6LocalServerStatistics 16	Number of DHCPv6 Renew packets received.
jnxJdhcpv6LocalServerRebindReceived	jnxJdhcpv6LocalServerStatistics 17	Number of DHCPv6 Rebind packets received.
jnxJdhcpv6LocalServerRelayForwReceived	jnxJdhcpv6LocalServerStatistics 18	Number of DHCPv6 Relay Forw packets received.
jnxJdhcpv6LocalServerRelayReplReceived	jnxJdhcpv6LocalServerStatistics 19	Number of DHCPv6 Relay Repl packets received.
jnxJdhcpv6LocalServerAdvertiseSent	jnxJdhcpv6LocalServerStatistics 20	Number of DHCPv6 Advertise packets sent.
jnxJdhcpv6LocalServerReplySent	jnxJdhcpv6LocalServerStatistics 21	Number of DHCPv6 Reply packets sent.
jnxJdhcpv6LocalServerReconfigureSent	jnxJdhcpv6LocalServerStatistics 22	Number of DHCPv6 Reconfigure packets sent.

The **jnxJdhcpv6LocalServerBindings**, whose object ID is **{jnxJdhcpv6LocalServerObjects 2}**, contains objects that describe the current statistics and bindings of a DHCPv6 local server.

The **jnxJdhcpv6LocalServerBindings** objects are listed in [Table 110 on page 399](#).

Table 110: jnxJdhcpv6LocalServerBindings

Object	Object Identifier	Description
jnxJdhcpv6LocalServerBindingsPrefix	jnxJdhcpv6LocalServerBindingsEntry 1	Prefix associated with this entry in the bindings table.
jnxJdhcpv6LocalServerBindingsLength	jnxJdhcpv6LocalServerBindingsEntry 2	Length of the prefix in bits.
jnxJdhcpv6LocalServerBindingsState	jnxJdhcpv6LocalServerBindingsEntry 3	State associated with this entry in the bindings table.
jnxJdhcpv6LocalServerBindingsLeaseEndTime	jnxJdhcpv6LocalServerBindingsEntry 4	Time the lease expires on this binding.
jnxJdhcpv6LocalServerBindingsLeaseExpireTime	jnxJdhcpv6LocalServerBindingsEntry 5	Time remaining until the lease expires for this binding.

Table 110: jnxJdhcpv6LocalServerBindings (*continued*)

Object	Object Identifier	Description
jnxJdhcpv6LocalServerBindingsLeaseStartTime	jnxJdhcpv6LocalServerBindingsEntry 6	Time the lease started for this binding.
jnxJdhcpv6LocalServerBindingsIncomingClientInterface	jnxJdhcpv6LocalServerBindingsEntry 7	Incoming interface for this binding.
jnxJdhcpv6LocalServerBindingsClientInterfaceVlanId	jnxJdhcpv6LocalServerBindingsEntry 8	VLAN ID for this binding.
jnxJdhcpv6LocalServerBindingsDemuxInterfaceName	jnxJdhcpv6LocalServerBindingsEntry 9	Demux interface for this binding.
jnxJdhcpv6LocalServerBindingsServerIpAddress	jnxJdhcpv6LocalServerBindingsEntry 10	IP Address associated with the server for this entry in the bindings table.
jnxJdhcpv6LocalServerBindingsBootpRelayAddress	jnxJdhcpv6LocalServerBindingsEntry 11	BOOTP relay address associated with the server for this entry in the bindings table.
jnxJdhcpv6LocalServerBindingsPreviousBootpRelayAddress	jnxJdhcpv6LocalServerBindingsEntry 12	Previous BOOTP relay address associated with the server for this entry in the bindings table.
jnxJdhcpv6LocalServerBindingsClientPoolName	jnxJdhcpv6LocalServerBindingsEntry 13	Client pool name.
jnxJdhcpv6LocalServerBindingsClientProfileName	jnxJdhcpv6LocalServerBindingsEntry 14	Client profile name.

The **jnxJdhcpv6LocalServerTraps**, whose object ID is **{jnxJdhcpv6LocalServerObjects 3}** contains objects that describe the DHCPv6 local server notifications.

The **jnxJdhcpv6LocalServerTraps** objects are listed in [“Juniper Networks Enterprise-Specific SNMP Traps” on page 57](#).

The **jnxJdhcpv6LocalServerTrapVars**, whose object ID is **{jnxJdhcpv6LocalServerObjects 4}**, contains objects that describe the current status and statistics of a DHCPv6 relay server.

The **jnxJdhcpv6LocalServerTrapVars** objects are listed in [Table 111 on page 401](#).

Table 111: jnxJdhcpv6LocalServerTrapVars

Object	Object Identifier	Description
jnxJdhcpv6LocalServerRouterName	jnxJdhcpv6LocalServerTrapVars 1	VRF ID in Junos OS. Represented as the Logical Router (LR) Name followed by the Router Instance (RI) Name.
jnxJdhcpv6LocalServerInterfaceName	jnxJdhcpv6LocalServerTrapVars 2	Interface where the DHCP client was detected.
jnxJdhcpv6LocalServerInterfaceLimit	jnxJdhcpv6LocalServerTrapVars 3	Number of clients supported on this interface.

jnxJdhcpv6LocalServerIfcStats, whose object ID is {jnxJdhcpv6LocalServerObjects 5}, contains objects that describe the contents of a table of interface statistics maintained by the DHCPv6 local server.

jnxJdhcpv6LocalServerIfcStats objects are listed in [Table 112 on page 401](#).

Table 112: jnxJdhcpv6LocalServerIfcStats

Object	Object Identifier	Description
jnxJdhcpv6LocalServerIfcStatsIfIndex	jnxJdhcpv6LocalServerIfcStatsEntry 1	IfIndex value of the interface for which this entry contains information.
jnxJdhcpv6LocalServerIfcStatsTotalDropped	jnxJdhcpv6LocalServerIfcStatsEntry 2	Total number of DHCPv6 packets dropped.
jnxJdhcpv6LocalServerIfcStatsNoSafdDropped	jnxJdhcpv6LocalServerIfcStatsEntry 3	Number of DHCPv6 packets dropped due to no safd match.
jnxJdhcpv6LocalServerIfcStatsBadSendDropped	jnxJdhcpv6LocalServerIfcStatsEntry 4	Number of DHCPv6 packets dropped due to send error.
jnxJdhcpv6LocalServerIfcStatsShortPacketDropped	jnxJdhcpv6LocalServerIfcStatsEntry 5	Number of DHCPv6 packets dropped due to packet being too short.
jnxJdhcpv6LocalServerIfcStatsBadMsgtypeDropped	jnxJdhcpv6LocalServerIfcStatsEntry 6	Number of DHCPv6 packets dropped due to bad options code in the packet.
jnxJdhcpv6LocalServerIfcStatsBadOptionsDropped	jnxJdhcpv6LocalServerIfcStatsEntry 7	Number of DHCPv6 packets dropped due to bad options in the packet.
jnxJdhcpv6LocalServerIfcStatsBadSrcAddressDropped	jnxJdhcpv6LocalServerIfcStatsEntry 8	Number of DHCPv6 packets dropped due to invalid address family.

Table 112: jnxJdhcpv6LocalServerIfcStats (*continued*)

Object	Object Identifier	Description
jnxJdhcpv6LocalServerIfcStatsRelayCountDropped	jnxJdhcpv6LocalServerIfcStatsEntry 9	Number of DHCPv6 packets dropped due to maximum relays supported.
jnxJdhcpv6LocalServerIfcStatsNoClientIdDropped	jnxJdhcpv6LocalServerIfcStatsEntry 10	Number of DHCPv6 packets dropped due to missing client ID.
jnxJdhcpv6LocalServerIfcStatsDeclineReceived	jnxJdhcpv6LocalServerIfcStatsEntry 11	Number of DHCPv6 Decline packets received.
jnxJdhcpv6LocalServerIfcStatsSolicitReceived	jnxJdhcpv6LocalServerIfcStatsEntry 12	Number of DHCPv6 Solicit packets received.
jnxJdhcpv6LocalServerIfcStatsInformationRequestReceived	jnxJdhcpv6LocalServerIfcStatsEntry 13	Number of DHCPv6 Information Request packets received.
jnxJdhcpv6LocalServerIfcStatsReleaseReceived	jnxJdhcpv6LocalServerIfcStatsEntry 14	Number of DHCPv6 Release packets received.
jnxJdhcpv6LocalServerIfcStatsRequestReceived	jnxJdhcpv6LocalServerIfcStatsEntry 15	Number of DHCPv6 Request packets received.
jnxJdhcpv6LocalServerIfcStatsConfirmReceived	jnxJdhcpv6LocalServerIfcStatsEntry 16	Number of DHCPv6 Confirm packets received.
jnxJdhcpv6LocalServerIfcStatsRenewReceived	jnxJdhcpv6LocalServerIfcStatsEntry 17	Number of DHCPv6 Renew packets received.
jnxJdhcpv6LocalServerIfcStatsRebindReceived	jnxJdhcpv6LocalServerIfcStatsEntry 18	Number of DHCPv6 Rebind packets received.
jnxJdhcpv6LocalServerIfcStatsRelayForwReceived	jnxJdhcpv6LocalServerIfcStatsEntry 19	Number of DHCPv6 Relay Forward packets received.
jnxJdhcpv6LocalServerIfcStatsRelayReplReceived	jnxJdhcpv6LocalServerIfcStatsEntry 20	Number of DHCPv6 Relay Reply packets received.
jnxJdhcpv6LocalServerIfcStatsAdvertiseSent	jnxJdhcpv6LocalServerIfcStatsEntry 21	Number of DHCPv6 Advertise packets sent.
jnxJdhcpv6LocalServerIfcStatsReplySent	jnxJdhcpv6LocalServerIfcStatsEntry 22	Number of DHCPv6 Reply packets sent.

Table 112: jnxJdhcpv6LocalServerIfcStats (*continued*)

Object	Object Identifier	Description
jnxJdhcpv6LocalServerIfcStatsReconfigureSent	jnxJdhcpv6LocalServerIfcStatsEntry 23	Number of DHCPv6 Reconfigure packets sent.
jnxJdhcpv6LocalServerIfcStatsTotalLeaseCount	jnxJdhcpv6LocalServerIfcStatsEntry 24	Number of Bound DHCP clients.
jnxJdhcpv6LocalServerIfcStatsStrictReconfigDropped	jnxJdhcpv6LocalServerIfcStatsEntry 25	Number of packets dropped due to strict reconfigure.
jnxJdhcpv6LocalServerIfcStatsAuthenticationDropped	jnxJdhcpv6LocalServerIfcStatsEntry 26	Number of packets dropped due to authentication failure.
jnxJdhcpv6LocalServerIfcStatsDynamicProfileDropped	jnxJdhcpv6LocalServerIfcStatsEntry 27	Number of packets dropped due to dynamic profile error.
jnxJdhcpv6LocalServerIfcStatsLicenseDropped	jnxJdhcpv6LocalServerIfcStatsEntry 28	Number of packets dropped due to license error.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 21

Digital Optical Monitoring MIB

- Interpreting the Enterprise-Specific Digital Optical Monitoring MIB on page 405

Interpreting the Enterprise-Specific Digital Optical Monitoring MIB

- Digital Optical Monitoring MIB on page 405
- DOM MIB Textual Conventions on page 405
- `jnxDomCurrentTable` on page 406
- `jnxDomNotifications` on page 408

Digital Optical Monitoring MIB

The Juniper Networks enterprise-specific Digital Optical Monitoring (DOM) MIB, whose object identifier is `{jnxDomMib1}`, under the existing **JUNIPER-SMI MIB** hierarchy, supports **SNMP Get** for statistics and **SNMP Trap** for alarms.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-dom.txt.

For information about the enterprise-specific DOM MIB objects, see the following topics:

- DOM MIB Textual Conventions on page 405
- `jnxDomCurrentTable` on page 406
- `jnxDomNotifications` on page 408

DOM MIB Textual Conventions

`jnxDomAlarmId` specifies the DOM alarms present on an interface. The possible values are:

- `domRxLossSignalAlarm(0)`—Input loss of signal
- `domRxCDRLossLockAlarm(1)`—Input loss of lock
- `domRxNotReadyAlarm(2)`—Input rx path
- `domRxLaserPowerHighAlarm(3)`—High input laser power
- `domRxLaserPowerLowAlarm(4)`—Low input laser power

- **domTxLaserBiasCurrentHighAlarm(5)**—High output laser bias current
- **domTxLaserBiasCurrentLowAlarm(6)**—Low output laser bias current
- **domTxLaserOutputPowerHighAlarm(7)**—High output laser power
- **domTxLaserOutputPowerLowAlarm(8)**—Low output laser power
- **domTxDataNotReadyAlarm(9)**—Output A/D data not ready
- **domTxNotReadyAlarm(10)**—Output tx path
- **domTxLaserFaultAlarm(11)**—Output laser safety
- **domTxCDRLossLockAlarm(12)**—Output clock data recovery
- **domModuleTemperatureHighAlarm(13)**—High module temperature
- **domModuleTemperatureLowAlarm(14)**—Low module temperature
- **domModuleNotReadyAlarm(15)**—Module not ready
- **domModulePowerDownAlarm(16)**—Module P_DOWN
- **domLinkDownAlarm(17)**—Wire unplugged or down
- **domModuleRemovedAlarm(18)**—Module unplugged or down

jnxDomWarningId specifies the DOM warnings present on an interface. The possible values are:

- **domRxLaserPowerHighWarning(0)**—High input laser power
- **domRxLaserPowerLowWarning(1)**—Low input laser power
- **domTxLaserBiasCurrentHighWarning(2)**—High output laser bias current
- **domTxLaserBiasCurrentLowWarning(3)**—Low output laser bias current
- **domTxLaserOutputPowerHighWarning(4)**—High output laser power
- **domTxLaserOutputPowerLowWarning(5)**—Low output laser power
- **domModuleTemperatureHighWarning(6)**—High module temperature
- **domModuleTemperatureLowWarning(7)**—Low module temperature

jnxDomCurrentTable

jnxDomCurrentTable is a table of **jnxDomCurrentEntry** objects that has DOM information about the router. **jnxDomCurrentEntry**, whose object ID is **{jnxDomCurrentTable 1}**, contains the objects listed in [Table 113 on page 406](#). Objects in the **jnxDomCurrentEntry** are indexed by **ifIndex**.

Table 113: jnxDomCurrentEntry

Object	Object ID	Description
jnxDomCurrentAlarms	jnxDomCurrentEntry 1	Identifies all the active DOM alarms on an XFP physical interface on the device.

Table 113: jnxDomCurrentEntry (*continued*)

Object	Object ID	Description
jnxDomCurrentAlarmDate	jnxDomCurrentEntry 2	Indicates the system date and time when the management subsystem learned of the current alarm.
jnxDomLastAlarms	jnxDomCurrentEntry 3	Identifies a copy of jnxDomCurrentAlarms before the last set or clear.
jnxDomCurrentWarnings	jnxDomCurrentEntry 4	Identifies all the active DOM warnings on an XFP physical interface on this device.
jnxDomCurrentRxLaserPower	jnxDomCurrentEntry 5	Indicates the receiver laser power in 0.01 dbm units.
jnxDomCurrentTxLaserBiasCurrent	jnxDomCurrentEntry 6	Indicates the receiver laser bias current in 0.001 mA units.
jnxDomCurrentTxLaserOutputPower	jnxDomCurrentEntry 7	Indicates the transmitter laser output power in 0.01 dbm units.
jnxDomCurrentModuleTemperature	jnxDomCurrentEntry 8	Indicates the module temperature in Celsius (degrees C).
jnxDomCurrentRxLaserPowerHighAlarmThreshold	jnxDomCurrentEntry 9	Indicates the receiver laser power high alarm threshold in 0.01 dbm units.
jnxDomCurrentRxLaserPowerLowAlarmThreshold	jnxDomCurrentEntry 10	Indicates the receiver laser power low alarm threshold in 0.01 dbm units.
jnxDomCurrentRxLaserPowerHighWarningThreshold	jnxDomCurrentEntry 11	Indicates the receiver laser power high warning threshold in 0.01 dbm units.
jnxDomCurrentRxLaserPowerLowWarningThreshold	jnxDomCurrentEntry 12	Indicates the receiver laser power low warning threshold in 0.01 dbm units.
jnxDomCurrentTxLaserBiasCurrentHighAlarmThreshold	jnxDomCurrentEntry 13	Indicates the transmitter laser bias current high alarm threshold in 0.001 mA units.
jnxDomCurrentTxLaserBiasCurrentLowAlarmThreshold	jnxDomCurrentEntry 14	Indicates the transmitter laser bias current low alarm threshold in 0.001 mA units.
jnxDomCurrentTxLaserBiasCurrentHighWarningThreshold	jnxDomCurrentEntry 15	Indicates the transmitter laser bias current high warning threshold in 0.001 mA units.
jnxDomCurrentTxLaserBiasCurrentLowWarningThreshold	jnxDomCurrentEntry 16	Indicates the transmitter laser bias current low warning threshold in 0.001 mA units.
jnxDomCurrentTxLaserOutputPowerHighAlarmThreshold	jnxDomCurrentEntry 17	Indicates the transmitter laser power high alarm threshold in 0.01 dbm units.

Table 113: jnxDomCurrentEntry (*continued*)

Object	Object ID	Description
jnxDomCurrentTxLaserOutputPowerLowAlarmThreshold	jnxDomCurrentEntry 18	Indicates the transmitter laser power low alarm threshold in 0.01 dbm units.
jnxDomCurrentTxLaserOutputPowerHighWarningThreshold	jnxDomCurrentEntry 19	Indicates the transmitter laser power high warning threshold in 0.01 dbm units.
jnxDomCurrentTxLaserOutputPowerLowWarningThreshold	jnxDomCurrentEntry 20	Indicates the transmitter laser power low warning threshold in 0.01 dbm units.
jnxDomCurrentModuleTemperatureHighAlarmThreshold	jnxDomCurrentEntry 21	Indicates the module temperature high alarm threshold in degrees Celsius.
jnxDomCurrentModuleTemperatureLowAlarmThreshold	jnxDomCurrentEntry 22	Indicates the module temperature low alarm threshold in degrees Celsius.
jnxDomCurrentModuleTemperatureHighWarningThreshold	jnxDomCurrentEntry 23	Indicates the module temperature high warning threshold in degrees Celsius.
jnxDomCurrentModuleTemperatureLowWarningThreshold	jnxDomCurrentEntry 24	Indicates the module temperature low warning threshold in degrees Celsius.

jnxDomNotifications

jnxDomNotifications contains information about notifications of recently set and cleared DOM alarms in the dense wavelength division multiplexing (DWDM) tunable 10-gigabit small form-factor pluggable (XFP) transceivers. **jnxDomAlarmSet** and **jnxDomAlarmCleared** are members of **jnxDomNotifications**.

jnxDomAlarmSet, whose object ID is {**jnxDomNotificationPrefix** 1}, contains information about a newly inserted DOM alarm condition on an interface, which is identified by **ifDescr**. There could be preexisting alarm conditions, which are indicated in **jnxDomLastAlarms**. The new alarm bit inserted is identified by comparing **jnxDomLastAlarms** and **jnxDomCurrentAlarms**. The time of this change is indicated in **jnxDomCurrentAlarmDate**. A change in the interface condition might trigger multiple **jnxDomAlarmSet** notifications. See [Table 114 on page 408](#).

Table 114: jnxDomAlarmSet

Object	Object ID	Description
ifDescr	ifEntry 2	Contains information about the interface.
jnxDomLastAlarms	jnxDomCurrentEntry 3	Contains preexisting alarm conditions.
jnxDomCurrentAlarms	jnxDomCurrentEntry 1	Contains active alarm conditions at the time when the new alarm bit was set.

Table 114: jnxDomAlarmSet (*continued*)

Object	Object ID	Description
jnxDomCurrentAlarmDate	jnxDomCurrentEntry 2	Contains the current date and time when the management subsystem learned of the current set DOM alarm.

jnxDomAlarmCleared, whose object ID is {**jnxDomNotificationPrefix 2**}, contains information about notifications when an existing DOM alarm condition is removed from an interface, which is identified by **ifDescr**. The clearing alarm bit can be identified by comparing **jnxDomLastAlarms** and **jnxDomCurrentAlarms**. The time of this change is indicated in **jnxDomCurrentAlarmDate**. A change in the interface condition might trigger multiple **jnxDomAlarmCleared** notifications. See [Table 115 on page 409](#).

Table 115: jnxDomAlarmCleared

Object	Object ID	Description
ifDescr	ifEntry 2	Contains information about the interface.
jnxDomLastAlarms	jnxDomCurrentEntry 3	Contains preexisting alarm conditions.
jnxDomCurrentAlarms	jnxDomCurrentEntry 1	Contains active alarm conditions at the time when a new alarm bit was cleared.
jnxDomCurrentAlarmDate	jnxDomCurrentEntry 2	Contains the current date and time when the management subsystem learned of the last cleared DOM alarm.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 22

DNS Objects MIB

- Interpreting the Enterprise-Specific DNS Objects MIB on page 411

Interpreting the Enterprise-Specific DNS Objects MIB

- DNS Objects MIB on page 411
- jnxJsDnsProxyDataObjects on page 411

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/junos13.2/topics/reference/mibs/mib-jnx-js-dns.txt.

For information about the enterprise-specific DNS Objects MIB, see the following topic:

- jnxJsDnsProxyDataObjects on page 411

jnxJsDnsProxyDataObjects

jnxJsDnsProxyDataObjects, whose object ID is **{jnxJsDns 1}**, displays the DNS query-related statistics listed in [Table 116 on page 411](#).

Table 116: 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.
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.

Table 116: jnxJsDnsProxyDataObjects (*continued*)

Object	Object ID	Description
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**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Dynamic Flow Capture MIB

- Interpreting the Enterprise-Specific Dynamic Flow Capture MIB on page 413

Interpreting the Enterprise-Specific Dynamic Flow Capture MIB

- Dynamic Flow Capture MIB on page 413
- jnxDfcCSTable on page 414
- jnxDfcCDTable on page 417
- DFC Notification Variables on page 418
- DFC Notification Definitions on page 419

Dynamic Flow Capture MIB

The Dynamic Flow Capture (DFC) PIC forwards passively monitored packets matching a particular filter list to one or more destinations.

The DFC architecture consists of one or more control sources that send requests to a Juniper Networks router to monitor incoming data and then forward any packets that match specific filter criteria to a set of one or more content destinations.

The Juniper Networks enterprise-specific DFC MIB, whose object identifier is **{jnxMibs 33}**, sends the current operational status for each DFC PIC. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-dfc.txt.



NOTE: The DFC PIC is supported on M320 and T320 routers and the T640 Core Router.

For information about the enterprise-specific DFC MIB objects, see the following topics:

- jnxDfcCSTable on page 414
- jnxDfcCDTable on page 417
- DFC Notification Variables on page 418
- DFC Notification Definitions on page 419

jnxDfcCSTable

The **jnxDfcCSTable**, whose object identifier is **{jnxDfc 1}**, provides information about the DFC control source on each DFC PIC on the router.

The **jnxDfcCSEntry**, whose object identifier is **{jnxDfcCSTable 1}**, has 44 objects, which are listed in [Table 117 on page 414](#).

Table 117: jnxDfcCSTable

Object	Object Identifier	Description
jnxDfcCSGrpName	jnxDfcCSEntry 1	The name assigned to a DFC group. A DFC group defines a profile of dynamic flow capture configuration information.
jnxDfcCSId	jnxDfcCSEntry 2	Control source identifier. The control source is a client that monitors electronic data or voice transfer over the network. The control source sends filter requests to the Juniper Networks router using a control protocol. The control source has a unique identifier and an optional list of IP addresses.
jnxDfcCSControlProtocolAddRequests	jnxDfcCSEntry 3	The number of control protocol Add requests received. The Add request specifies new filter criteria to be included in the current filter configuration for a given control source and content destination.
jnxDfcCSCriteriaAdded	jnxDfcCSEntry 4	The number of filter criteria added successfully by the control source.
jnxDfcCSCriteriaAdditionFailed	jnxDfcCSEntry 5	The number of filter criteria Add requests that could not be processed successfully.
jnxDfcCSControlProtocolDeleteRequests	jnxDfcCSEntry 6	The number of control protocol Delete requests received. The Delete request specifies filter criteria to be removed from the current filter configuration for a given control source and content destination.
jnxDfcCSCriteriaDeleted	jnxDfcCSEntry 7	The number of filter criteria deleted successfully as requested by the control source.
jnxDfcCSCriteriaDeletionFailed	jnxDfcCSEntry 8	The number of filter criteria Delete requests that could not be processed successfully.
jnxDfcCSCriteriaDeletedTimeoutIdle	jnxDfcCSEntry 9	The number of criteria deleted by timeout idle.
jnxDfcCSCriteriaDeletedTimeoutTotal	jnxDfcCSEntry 10	The number of criteria deleted by timeout total.
jnxDfcCSCriteriaDeletedPackets	jnxDfcCSEntry 11	The number of criteria deleted by packets.
jnxDfcCSCriteriaDeletedBytes	jnxDfcCSEntry 12	The number of criteria deleted by bytes.

Table 117: jnxDfcCSTable (*continued*)

Object	Object Identifier	Description
jnxDfcCSControlProtocolRefreshRequests	jnxDfcCSEntry 13	The number of control protocol Refresh requests received. The Refresh request updates the timeout for a particular filter criterion (or a set of filter criteria) for a given control source and content destination.
jnxDfcCSCriteriaRefreshed	jnxDfcCSEntry 14	The number of filter criteria Refresh requests processed successfully as requested by the control source.
jnxDfcCSCriteriaRefreshFailed	jnxDfcCSEntry 15	The number of filter criteria Refresh requests that could not be processed successfully.
jnxDfcCSControlProtocolListRequests	jnxDfcCSEntry 16	The number of control protocol List requests received. The List request returns a list of all criteria that a particular control source has added and are currently active.
jnxDfcCSListSuccess	jnxDfcCSEntry 17	The number of List requests processed successfully as requested by the control source.
jnxDfcCSListFailed	jnxDfcCSEntry 18	The number of List requests that could not be processed successfully.
jnxDfcCSControlProtocolNoopRequests	jnxDfcCSEntry 19	The number of control protocol Noop requests received. This request is used to verify the end-to-end connectivity between the control source and the DFC PIC.
jnxDfcCSNoopSuccess	jnxDfcCSEntry 20	The number of Noop requests processed successfully as requested by the control source.
jnxDfcCSNoopFailed	jnxDfcCSEntry 21	The number of Noop requests that could not be processed successfully.
jnxDfcCSDynamicCriteriaActive	jnxDfcCSEntry 22	The number of active dynamic filter criteria.
jnxDfcCSStaticCriteriaActive	jnxDfcCSEntry 23	The number of active static filter criteria.
jnxDfcCSBadRequest	jnxDfcCSEntry 24	The number of Bad requests received.
jnxDfcCSResponseSuccessful	jnxDfcCSEntry 25	The number of successful responses corresponding to the Add , Delete , Refresh , List , and Noop requests sent to the control source.
jnxDfcCSResponseImproperCriteria	jnxDfcCSEntry 26	The number of responses generated because of improper filter criteria included in an Add request.
jnxDfcCSResponseUnknownContentDest	jnxDfcCSEntry 27	The number of responses generated because of an unknown content destination included in an Add , Delete , Refresh , or List request.

Table 117: jnxDfcCSTable (*continued*)

Object	Object Identifier	Description
jnxDfcCSResponseUnknownControlSrc	jnxDfcCSEntry 28	The number of responses generated because of an unknown control source included in an Add , Delete , Refresh , or List request.
jnxDfcCSResponseUnknownCriteriaId	jnxDfcCSEntry 29	The number of responses generated because of an unknown criteria identifier included in an Add , Delete , Refresh , or List request.
jnxDfcCSResponseImproperTimeout	jnxDfcCSEntry 30	The number of responses generated because of an improper timeout specified in an Add or Refresh request.
jnxDfcCSResponseInvalidAuthentication	jnxDfcCSEntry 31	The number of responses generated because of invalid authentication information included in an Add , Delete , Refresh , List , or Noop request.
jnxDfcCSResponseInvalidSequenceNumber	jnxDfcCSEntry 32	The number of responses generated because of an invalid sequence number included in an Add , Delete , Refresh , List , or Noop request.
jnxDfcCSResponseInternalError	jnxDfcCSEntry 33	The number of responses generated because an internal error occurred on the DFC PIC processing the request.
jnxDfcCSNotificationRestart	jnxDfcCSEntry 34	The number of Restart notifications sent to configured notification recipients. A notification is generated when a system failure occurs and all DFC filter criteria are lost.
jnxDfcCSNotificationRollover	jnxDfcCSEntry 35	The number of Rollover notifications sent to configured notification recipients. A notification is generated when a sequence number rollover occurs on the DFC PIC.
jnxDfcCSNotificationNoop	jnxDfcCSEntry 36	The number of Noop notifications sent to configured notification recipients. A notification is generated when the DFC PIC receives a Noop message that includes a SendAsync parameter.
jnxDfcCSNotificationTimeout	jnxDfcCSEntry 37	The number of Timeout notifications sent to configured notification recipients. This notification is generated when a DFC PIC times out a filter criterion (based on any one of its configured timeout parameters) and the criterion contains a SendTimeoutAsync parameter.
jnxDfcCSNotificationCongestion	jnxDfcCSEntry 38	A Congestion notification is generated when the total 10-second average packet forwarding rate (in bps) summed over all active filter criteria to a configured content destination exceeds the configured <i>soft</i> limit for the destination. The jnxDfcCSNotificationCongestion object contains the number of Congestion notifications sent to configured notification recipients.

Table 117: jnxDfcCSTable (*continued*)

Object	Object Identifier	Description
jnxDfcCSNotificationCongestionDelete	jnxDfcCSEntry 39	A Congestion Delete notification is generated when the total 10-second average packet forwarding rate (in bps) summed over all active filter criteria to a configured content destination exceeds the configured <i>hard</i> limit for the destination. The jnxDfcCSNotificationCongestionDelete object contains the number of Congestion Delete notifications sent to configured notification recipients.
jnxDfcCSNotificationDuplicatesDropped	jnxDfcCSEntry 40	The number of Duplicated Dropped notifications sent to configured notification recipients. This notification is generated when the configurable Maximum Duplicates parameter has been exceeded and packets matching criteria added by the corresponding control source are dropped.
jnxDfcCSAddRequestRate	jnxDfcCSEntry 41	The request processing rate (in requests processed per second).
jnxDfcCSAddRequestPeakRate	jnxDfcCSEntry 42	The peak request processing rate (in requests processed per second).
jnxDfcCSAggrCriteriaBandwidth	jnxDfcCSEntry 43	Bandwidth (in bps).
jnxDfcCSSequenceNumber	jnxDfcCSEntry 44	Protocol sequence number.

jnxDfcCDTable

The **jnxDfcCDTable**, whose object identifier is **{jnxDfc 2}**, provides statistical information for content destinations.

The **jnxDfcCDEntry**, whose object identifier is **{jnxDfcCDTable 1}**, has seven objects, which are listed in [Table 118 on page 417](#).

Table 118: jnxDfcCDTable

Object	Object Identifier	Description
jnxDfcCDGrpName	jnxDfcCDEntry 1	The name assigned to a DFC group. A DFC group defines a profile of dynamic flow capture configuration information.
jnxDfcCDId	jnxDfcCDEntry 2	Content destination identifier. The DFC router processes the requests from the control sources, creates the filters, monitors incoming data flows, and sends the matched packets to their respective content destinations. Content destinations receive the matched packets from the router.
jnxDfcCDCriteria	jnxDfcCDEntry 3	The number of filter criteria configured for the content destination.
jnxDfcCDByteRate	jnxDfcCDEntry 4	The average data rate (in bytes per second) summed over all active filter criteria configured for a given content destination.

Table 118: jnxDfcCDTable (*continued*)

Object	Object Identifier	Description
jnxDfcCDMatchedPackets	jnxDfcCDEntry 5	The number of packets that match the filter criteria configured for a content destination.
jnxDfcCDMatchedBytes	jnxDfcCDEntry 6	The number of bytes that match the filter criteria configured for a content destination.
jnxDfcCDCongestionNotification	jnxDfcCDEntry 7	The number of Congestion notifications sent to a configured notification recipient.

DFC Notification Variables

The enterprise-specific DFC MIB provides notifications for monitoring dynamic flow capture. [Table 119 on page 418](#) lists the supported notification variables.

Table 119: Supported Notification Variables for the DFC MIB

Object	Object Identifier	Description
jnxDfcInputPktRate	jnxDfcNotifyVars 1	Data packet rate (in pps).
jnxDfcPpsSoftOverloadLowWatermark	jnxDfcNotifyVars 2	Configured lowest value for the data packet rate (in pps).
jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotifyVars 3	Configured highest value for the data packet rate (in pps).
jnxDfcPpsHardOverloadLowWatermark	jnxDfcNotifyVars 4	Recommended lowest value for the data packet rate (in pps).
jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotifyVars 5	Recommended highest value for the data packet rate (in pps).
jnxDfcFlowsUsage	jnxDfcNotifyVars 6	Percent (%) usage of the total number of flows.
jnxDfcCriteriaUsage	jnxDfcNotifyVars 7	Percent (%) usage of matching criteria for all filters.
jnxDfcMemSoftOverloadLowWatermark	jnxDfcNotifyVars 8	Configured lowest watermark percent for memory load.
jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotifyVars 9	Configured highest watermark percent for memory load.
jnxDfcFlowLowWatermark	jnxDfcNotifyVars 10	Recommended lowest value for the number of flows allowed.
jnxDfcFlowHighWatermark	jnxDfcNotifyVars 11	Recommended highest value for the number of flows allowed.
jnxDfcCriteriaLowWatermark	jnxDfcNotifyVars 12	Recommended lowest value for the number of criteria allowed.
jnxDfcCriteriaHighWatermark	jnxDfcNotifyVars 13	Recommended highest value for the number of criteria allowed.

DFC Notification Definitions

Table 120 on page 419 lists the supported notification definitions.

Table 120: Supported Notification Definitions for the DFC MIB

Notification Type	Objects	Identifier	Description
jnxDfcSoftPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 1	Notification that occurs when the input packet rate (in pps) exceeds the configured limit.
jnxDfcSoftPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 2	Notification that occurs when the input packet rate (in pps) returns to below the configured limit.
jnxDfcHardPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 3	Notification that occurs when the input packet rate (in pps) exceeds the recommended limit.
jnxDfcHardPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 4	Notification that occurs when the input packet rate (in pps) returns to below the recommended limit.
jnxDfcSoftMemThresholdExceeded	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 5	Notification that occurs when memory usage exceeds the configured limit.
jnxDfcSoftMemUnderThreshold	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 6	Notification that occurs when memory usage returns to below the configured limit.
jnxDfcHardMemThresholdExceeded	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 7	Notification that occurs when memory usage exceeds the recommended limit.
jnxDfcHardMemUnderThreshold	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 8	Notification that occurs when memory usage returns to below the recommended limit.

Related Documentation • [Standard SNMP MIBs Supported by Junos OS on page 9](#)

- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 24

Ethernet MAC MIB

- [Interpreting the Enterprise-Specific Ethernet MAC MIB on page 421](#)

Interpreting the Enterprise-Specific Ethernet MAC MIB

- [Ethernet MAC MIB on page 421](#)
- [jnxMacStatsTable on page 421](#)

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/junos13.2/topics/reference/mibs/mib-jnx-mac.txt.

For information about the enterprise-specific Ethernet MAC MIB objects, see the following topic:

- [jnxMacStatsTable on page 421](#)

jnxMacStatsTable

The **jnxMacStatsTable** contains a list of MAC statistics for Gigabit Ethernet interfaces.

jnxMacStatsEntry objects are listed in [Table 121 on page 421](#).

Table 121: jnxMacStatsTable

Object	Object Identifier	Description
jnxVlanIndex	jnxMacStatsEntry 1	The virtual LAN (VLAN) ID of a VLAN.
jnxSourceMacAddress	jnxMacStatsEntry 2	The source MAC address.
jnxMacHCInOctets	jnxMacStatsEntry 3	The number of total octets received in this VLAN/MAC address.
jnxMacHCInFrames	jnxMacStatsEntry 4	The number of total frames received in this VLAN/MAC address

Table 121: jnxMacStatsTable (*continued*)

Object	Object Identifier	Description
jnxMacHCOctets	jnxMacStatsEntry 5	The number of total octets transmitted in this VLAN/MAC address.
jnxMacHCOFrames	jnxMacStatsEntry 6	The number of total frames transmitted in this VLAN/MAC address.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 25

Event MIB

- Interpreting the Enterprise-Specific Event MIB on page 423

Interpreting the Enterprise-Specific Event MIB

- Event MIB on page 423
- jnxEventAvTable on page 423
- Notifications for the Event MIB on page 424

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/junos13.2/topics/reference/mibs/mib-jnx-event.txt.

For information about the enterprise-specific Event MIB objects, see the following topics:

- jnxEventAvTable on page 423
- Notifications for the Event MIB on page 424

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 122 on page 423](#).

Table 122: 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.

Table 122: jnxEventAvTable (*continued*)

Object	Object Identifier	Description
jnxEventAvAttribute	jnxEventAvEntry 2	The attribute name in the trap generated by an op script or event policy.
jnxEventAvValue	jnxEventAvEntry 3	The value of the attribute identified by jnxEventAvAttribute .

Notifications for the Event MIB

Table 123 on page 424 lists the supported notifications for the Event MIB.

Table 123: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 26

Experiment MIB

- [jnxExperiment on page 425](#)

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 819](#)
- [jnxProducts on page 819](#)
- [jnxServices on page 820](#)
- [jnxMibs on page 821](#)
- [jnxTraps on page 822](#)

CHAPTER 27

EX Series MAC Notification MIB

- [Interpreting the Enterprise-Specific EX Series MAC Notification MIB on page 427](#)

Interpreting the Enterprise-Specific EX Series MAC Notification MIB

- [EX Series MAC Notification MIB on page 427](#)
- [MAC Notification Global Group on page 427](#)
- [MAC Notification MIB Global Objects on page 428](#)
- [MAC History Index on page 429](#)
- [MAC History Changes Message on page 429](#)

EX Series MAC Notification MIB

The **jnxMacNotification** MIB module, whose object ID is **{jnxExSwitching 7}**, is for configuring the MAC notification feature on EX Series Ethernet Switches. MAC notification is a mechanism that is used to inform monitoring devices when there are MAC addresses learned or removed from the forwarding database of the monitored devices.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ex-mac-notification.txt.

For information about the Juniper Networks enterprise-specific EX Series MAC Notification MIB objects, see the following topics:

- [MAC Notification Global Group on page 427](#)
- [MAC Notification MIB Global Objects on page 428](#)
- [MAC History Index on page 429](#)
- [MAC History Changes Message on page 429](#)

MAC Notification Global Group

The **jnxMacGlobalFeatureEnabled** object indicates whether the MAC notification feature is currently running in the device. To disable the MAC notification feature globally, set this object to false. The feature is disabled at each interface.

Set this object to true to start the MAC notification feature running on the device. Once the MAC notification is enabled, the **jnxIfConfigTable** controls the functioning of the feature.

MAC Notification MIB Global Objects

jnxMacNotificationMIBGlobalObjects table, whose object identifier is **{jnxMacNotificationMIBGlobalObjects 1}**, contains details about MAC notifications. Each **jnxMacNotification** contains the objects listed in [Table 124 on page 428](#).

Table 124: jnxMacNotificationGlobalObjects

Object	Object ID	Description
jnxMacNotificationInterval	jnxMacNotificationMIBGlobalObjects 1	The MAC notification interval object specifies the maximum interval of time between notifications that are generated by the device. If the value of jnxNotificationsEnabled is true, the device sends out the generated notifications and archives the MAC change notification events in the jnxMacHistory table. If the value of jnxNotificationEnabled is false, the device does not send out the generated notifications, but it will archive these events in the jnxMacHistory table.
jnxMacAddressesLearnt	jnxMacNotificationMIBGlobalObjects 2	Indicates the number of MAC addresses learned by the device.
jnxMacAddressesRemoved	jnxMacNotificationMIBGlobalObjects 3	Indicates the number of MAC addresses removed from the forwarding database.
jnxMacNotificationsEnabled	jnxMacNotificationMIBGlobalObjects 4	Indicates whether MAC notifications will or will not be sent when there are MAC addresses learned or removed from the device's forwarding database. Disabling notifications does not prevent the MAC address information from being added to the jnxMacHistoryTable .
jnxMacNotificationsSent	jnxMacNotificationMIBGlobalObjects 5	Indicates the number of MAC notifications sent out by the device.
jnxMacHistTableMaxLength	jnxMacNotificationMIBGlobalObjects 6	Indicates the upper limit on the number of entries that the jnxMacHistoryTable can contain. A value of 0 prevents any history from being retained. When the table is full, the oldest entry is deleted and a new entry is created.
jnxMacHistoryTable	jnxMacNotificationMIBGlobalObjects 7	Archives the MAC change notification events generated by the device. The MAC change notification events are archived here even if jnxMacChangesNotifications are not actually sent.
jnxMacHistoryEntry	jnxMacNotificationMIBGlobalObjects 8	A MAC change notification message that was previously generated by the device. Each entry is indexed by a message index.

MAC History Index

jnxHistIndex is an index that uniquely identifies a MAC change notification event previously generated by the device. This index starts at 1 and increases by one when a MAC change notification is generated. When it reaches the maximum value (**4294967295**), the agent wraps the value back to 1.

MAC History Changes Message

The **jnxHistMacChangedMsg** object contains details of a MAC change notification event. It consists of several tuples packed together in the format **tuple1, tuple2, ...**.

Each tuple comprises 13 octets in the following format:

<operation><vlan><MAC><dot1dBasePort>.

<operation> is of size 1 octet and supports the following values:

- 0—End of MIB object.
- 1—MAC learned.
- 2—MAC removed.

<vlan> is the VLAN number of the VLAN whose MAC address is being used and has a size of 2 octets.

<Mac> is the Layer 2 MAC address and has a size of 6 octets.

<dot1dBasePort> is the value of **dot1dBasePort** for the interface from which the MAC address is learned and has a size of 4 octets.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 28

EX Series SMI MIB

- Interpreting the Enterprise-Specific EX Series SMI MIB on page 431

Interpreting the Enterprise-Specific EX Series SMI MIB

- EX Series SMI MIB on page 431

EX Series SMI MIB

The Juniper Networks enterprise-specific EX Series Structure of Management Information (SMI) MIB leverages the **jnxExMibRoot** object from the enterprise-specific SMI MIB (http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-smi.txt), and defines a MIB branch for switching-related MIB definitions for the EX Series Ethernet Switches. MIB objects that are specific to EX Series are identified with a **jnxEx** prefix.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ex-smi.txt.

The **jnxExMibRoot** contains one branch, **jnxExSwitching**, whose object identifier is **{jnxExMibRoot 1}**.

The **jnxExSwitching** branch contains the objects listed in [Table 125 on page 431](#).

Table 125: jnxExSwitching

Object	Object ID	Description
jnxExAnalyzer	jnxExSwitching 1	Defines the branch for the enterprise-specific Analyzer MIB. For more information about the Analyzer MIB, see “ Analyzer MIB ” on page 81.
jnxExSecureAccessPort	jnxExSwitching 2	Defines the branch for the enterprise-specific Secure Access Port MIB. NOTE: The Secure Access Port MIB is not currently supported on EX Series Switches.
jnxExPaeExtention	jnxExSwitching 3	Defines the branch for the enterprise-specific PAE Extensions MIB. For more information about the PAE Extensions MIB, see “ PAE Extension MIB ” on page 655.

Table 125: jnxExSwitching (*continued*)

Object	Object ID	Description
jnxExVirtualChassis	jnxExSwitching 4	Defines the branch for the enterprise-specific Virtual Chassis MIB. For more information about the Virtual Chassis MIB, see “Virtual Chassis MIBs” on page 433 .
jnxExVlan	jnxExSwitching 5	Defines the branch for the enterprise-specific VLAN MIB. For more information about the VLAN MIB, see “VLAN MIB” on page 841 .

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

EX Series Virtual Chassis MIB

- [Interpreting the Enterprise-Specific Virtual Chassis MIB on page 433](#)

Interpreting the Enterprise-Specific Virtual Chassis MIB

- [Virtual Chassis MIBs on page 433](#)
- [jnxVirtualChassisMemberTable on page 433](#)

Virtual Chassis MIBs

The Juniper Networks enterprise-specific Virtual Chassis MIB, whose object identifier (OID) is {jnxExSwitching 4}, contains information about virtual chassis on EX Series Ethernet Switches and the MX Series.



NOTE: You have to get the MIB objects by using the member ID and the port name as it is currently done on EX Series Switches and cannot use the SNMP index as it is done for network interfaces.

For information about the enterprise-specific Virtual Chassis MIB objects, see the following topics:

- [jnxVirtualChassisMemberTable on page 433](#)
- [*jnxVirtualChassisPortTable*](#)

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-virtualchassis.txt.

jnxVirtualChassisMemberTable

The jnxVirtualChassisMemberTable enterprise-specific Virtual Chassis MIB, whose object identifier is {jnxVirtualChassisMemberMIB 1}, contains information about the switches that form a Virtual Chassis. Each jnxVirtualChassisMemberEntry contains the objects listed in [Table 126 on page 434](#).

Table 126: jnxVirtualChassisMemberTable

Object	Object ID	Description
jnxVirtualChassisMemberId	jnxVirtualChassisMemberEntry 1	Uniquely identifies a Virtual chassis member. This object contains integer values in the range 0 through 9.
jnxVirtualChassisMemberSerialnumber	jnxVirtualChassisMemberEntry 2	Contains the serial number of the Virtual Chassis member.
jnxVirtualChassisMemberRole	jnxVirtualChassisMemberEntry 3	Specifies the role of Virtual Chassis member. This object uses the following integer values: <ul style="list-style-type: none"> • 1—Master • 2—Backup • 3—Linecard
jnxVirtualChassisMemberMacAddBase	jnxVirtualChassisMemberEntry 4	Specifies the media access control (MAC) address base for the virtual-chassis member.
jnxVirtualChassisMemberSWVersion	jnxVirtualChassisMemberEntry 5	Identifies the Junos OS version that is installed on the Virtual Chassis member.
jnxVirtualChassisMemberPriority	jnxVirtualChassisMemberEntry 6	Specifies the mastership priority for the Virtual Chassis member. This object contains integer values in the range 1 through 255.
jnxVirtualChassisMemberUptime	jnxVirtualChassisMemberEntry 7	Specifies the uptime information for a Virtual Chassis member.

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 30

Firewall MIB

- Interpreting the Enterprise-Specific Firewall MIB on page 435

Interpreting the Enterprise-Specific Firewall MIB

- Firewall MIB on page 435
- `jnxFirewallsTable` on page 436
- `jnxFirewallCounterTable` on page 436

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/junos13.2/topics/reference/mibs/mib-jnx-firewall.txt.

For information about the enterprise-specific Firewall MIB objects, see the following topics:

- `jnxFirewallsTable` on page 436
- `jnxFirewallCounterTable` on page 436

jnxFirewallsTable

The deprecated **jnxFirewallsTable** contains **jnxFirewallsEntry**, whose object ID is **{jnxFirewallsTable 1}**. Each **jnxFirewallsEntry** contains the objects listed in [Table 127 on page 436](#).

Table 127: 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.

jnxFirewallCounterTable

The **jnxFirewallCounterTable**, whose object identifier is **{jnxFirewalls 2}**, replaces the deprecated **jnxFirewallsTable**. Each **jnxFirewallCounterEntry** contains the objects listed in [Table 128 on page 436](#).

Table 128: jnxFirewallCounterEntry

Object	Object ID	Description
jnxFWCounterFilterName	jnxFirewallCounterEntry 1	The name of the firewall filter. The name can have up to 127 characters.
jnxFWCounterName	jnxFirewallCounterEntry 2	The name of the counter or the policer. The name can have up to 127 characters.
jnxFWCounterType	jnxFirewallCounterEntry 3	The type of the jnxFWCounterName object. The value of jnxFWType can be 1 (other), 2 (counter), or 3 (policer).
jnxFWCounterPacketCount	jnxFirewallCounterEntry 4	The number of packets that are associated with the specified counter or policer.
jnxFWCounterByteCount	jnxFirewallCounterEntry 5	The number of bytes that are associated with the counter or policer.

Table 128: jnxFirewallCounterEntry (*continued*)

Object	Object ID	Description
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**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 31

Flow Collection Services MIB

- Interpreting the Enterprise-Specific Flow Collection Services MIB on page 439

Interpreting the Enterprise-Specific Flow Collection Services MIB

- Flow Collection Services MIB on page 439
- jnxCollGlobalStats on page 439
- jnxCollPicIfTable on page 440
- jnxCollFileTable on page 442

Flow Collection Services MIB

The Juniper Networks enterprise-specific Flow Collection Services MIB, whose object identifier is **{jnxMibs 28}**, provides statistics on files, records, memory, FTP, and error states of flow collection services on a Monitoring Services PIC. 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/junos13.2/topics/reference/mibs/mib-jnx-coll.txt.

For information about how to configure the flow collection services interface, see the *Junos OS Network Interfaces Library for Routing Devices* and the *Junos OS Feature Guide*.

For information about the enterprise-specific Flow Collection Services MIB objects, see the following topics:

- jnxCollGlobalStats on page 439
- jnxCollPicIfTable on page 440
- jnxCollFileTable on page 442

jnxCollGlobalStats

jnxCollGlobalStats provides statistics on all the router's Monitoring Services PICs and has the objects listed in [Table 129 on page 440](#).

Table 129: jnxCollGlobalStats

Object	Object Identifier	Description
jnxCollGlobalCreatedFiles	jnxCollGlobalStats 1	The number of files created by all the Monitoring Services PICs on the router since the last time the router was reset.
jnxCollGlobalOpenFiles	jnxCollGlobalStats 2	The number of open flow collection services files currently on the router.

jnxCollPicIfTable

jnxCollPicIfTable contains statistics about each Monitoring Services PIC.

JnxCollPicIfEntry contains information about a Monitoring Services PIC. Each Monitoring Services PIC contains one interface and is identified by **IfIndex**. It has objects listed in [Table 130 on page 440](#).

Table 130: jnxCollPicIfEntry

Object	Object Identifier	Description
File Statistics		
jnxCollPicIfCreatedFiles	jnxCollPicIfEntry 1	The number of files created by a Monitoring Services PIC since the last time the PIC was reset.
jnxCollPicIfCreatedFileRate	jnxCollPicIfEntry 2	The number of files created per second during the current 10-second interval.
jnxCollPicIfPeakCreatedFileRate	jnxCollPicIfEntry 3	The peak number of files created per second.
jnxCollPicIfExportedFiles	jnxCollPicIfEntry 4	The number of files exported by a Monitoring Services PIC.
jnxCollPicIfExportedFileRate	jnxCollPicIfEntry 5	The number of files exported per second during the current 10-second interval.
jnxCollPicIfPeakExportedFileRate	jnxCollPicIfEntry 6	The peak number of files exported per second.
jnxCollPicIfDestroyedFiles	jnxCollPicIfEntry 7	The number of files successfully exported and files dropped by the Monitoring Services PIC. Files are destroyed after they are transferred to the FTP server or when there is not enough memory.
jnxCollPicIfDestroyedFileRate	jnxCollPicIfEntry 8	The number of files dropped per second during the current 10-second interval. Files are dropped after they are transferred to the FTP server or when there is not enough memory.
jnxCollPicIfPeakDestroyedFileRate	jnxCollPicIfEntry 9	The peak number of files dropped, per second. Files are dropped after they are transferred to the FTP server or when there is not enough memory.
Record Statistics		
jnxCollPicIfProcRecords	jnxCollPicIfEntry 10	The number of flow records processed by a Monitoring Services PIC.

Table 130: jnxCollPicIfEntry (*continued*)

Object	Object Identifier	Description
jnxCollPicIfProcRecordsRate	jnxCollPicIfEntry 11	The number of flow records processed per second during the current 10-second interval.
jnxCollPicIfPeakProcRecordsRate	jnxCollPicIfEntry 12	The peak number of flow records processed.
Memory Statistics		
jnxCollPicIfMemoryUsed	jnxCollPicIfEntry 13	The amount of memory used, in bytes, by a Monitoring Services PIC.
jnxCollPicIfMemoryFree	jnxCollPicIfEntry 14	The amount of free memory, in bytes, on a Monitoring Services PIC.
FTP Statistics		
jnxCollPicIfFtpBytes	jnxCollPicIfEntry 15	The number of bytes transferred by a Monitoring Services PIC using FTP.
jnxCollPicIfFtpByteRate	jnxCollPicIfEntry 16	The number of bytes per second transferred using FTP, measured during the current 10-second interval.
jnxCollPicIfPeakFtpByteRate	jnxCollPicIfEntry 17	The peak number of bytes per second transferred using FTP.
jnxCollPicIfFtpFiles	jnxCollPicIfEntry 18	The number of files transferred by a Monitoring Services PIC using FTP.
jnxCollPicIfFtpFileRate	jnxCollPicIfEntry 19	The number of files per second transferred using FTP.
jnxCollPicIfPeakFtpFileRate	jnxCollPicIfEntry 20	The peak number of files per second transferred using FTP.
jnxCollPicIfFtpFailures	jnxCollPicIfEntry 21	The number of FTP transfer failures transferred by a Monitoring Services PIC.
Error State Statistics		
jnxCollPicIfCurrentState	jnxCollPicIfEntry 22	The current state of various error conditions on a Monitoring Services PIC.
jnxCollPicIfLastStateChange	jnxCollPicIfEntry 23	The error condition of the last changed state.
jnxCollPicIfStateChangeTime	jnxCollPicIfEntry 24	The value of sysUpTime when the management subsystem last learned of a change to the jnxCollPicIfCurrentState for a Monitoring Services PIC.
jnxCollPicIfStateChangeDate	jnxCollPicIfEntry 25	The system date and time when the management subsystem last learned of a change to the jnxCollPicIfCurrentState on a Monitoring Services PIC.
jnxCollPicIfStateChangeType	jnxCollPicIfEntry 26	Indicates whether the last state change set a new error condition or cleared an existing one. This object contains the following values: <ul style="list-style-type: none"> • none(1) • set(2) • cleared(3)

jnxCollFileTable

jnxCollFileTable contains information about each flow collection services file on the router.

jnxCollFileEntry contains information about a single file open on a Monitoring Services PIC, and has the objects listed in [Table 131 on page 442](#).

Table 131: jnxCollFileTable

Object	Object Identifier	Description
jnxCollFileName	jnxCollFileEntry 1	The name of a flow collection services file on a Monitoring Services PIC.
jnxCollFileFname	jnxCollFileEntry 2	The name of a flow collection services file on this Monitoring Services PIC. This object is included for those network management applications that cannot parse the filename from the instance portion of the OIDs and provides the value of jnxCollFileName .
jnxCollFileRecords	jnxCollFileEntry 3	The number of flow records in this file.
jnxCollFileRecordRate	jnxCollFileEntry 4	The number of flow records per second added to this file, measured during the current 10-second interval.
jnxCollFilePeakRecordRate	jnxCollFileEntry 5	The peak number of flow records per second added to this file.
jnxCollFileUncompBytes	jnxCollFileEntry 6	The number of uncompressed bytes in this file.
jnxCollFileUncompByteRate	jnxCollFileEntry 7	The number of uncompressed bytes per second added to this file.
jnxCollFilePeakUncompByteRate	jnxCollFileEntry 8	The peak number of uncompressed bytes per second added to this file.
jnxCollFileCompBytes	jnxCollFileEntry 9	The number of compressed bytes in this file.
jnxCollFileCompByteRate	jnxCollFileEntry 10	The number of compressed bytes per second added to this file during the current 10-second interval.
jnxCollFilePeakCompByteRate	jnxCollFileEntry 11	The peak number of compressed bytes per second added to this file.
jnxCollFileBlocks	jnxCollFileEntry 12	The number of blocks in this file.
jnxCollFileCompBlocks	jnxCollFileEntry 14	The number of compressed blocks in this file.
jnxCollFileTransferAttempts	jnxCollFileEntry 15	The number of FTP transfer attempts in this file.
jnxCollFileState	jnxCollFileEntry 16	<p>The current state of this file. This object contains the following values:</p> <ul style="list-style-type: none"> unknown(1) active(2)—The file is actively receiving flow records. wait(3)—The file is waiting for export. export1(4)—The file is being exported to the primary server. export2(5)—The file is being exported to the secondary server.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 32

Host Resources MIB

- [Interpreting the Enterprise-Specific Host Resources MIB on page 445](#)

Interpreting the Enterprise-Specific Host Resources MIB

- [Host Resources MIB on page 445](#)
- [jnxHrStorageTable on page 445](#)

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/junos13.2/topics/reference/mibs/mib-jnx-hostresources.txt.

For information about the enterprise-specific Host Resources MIB objects, see the following topic:

- [jnxHrStorageTable on page 445](#)

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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 33

Interface MIB

- [Interpreting the Enterprise-Specific Interface MIB on page 447](#)

Interpreting the Enterprise-Specific Interface MIB

- [Interface MIB on page 447](#)
- [jnxIfTable on page 447](#)
- [ifChassisTable on page 450](#)

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/junos13.2/topics/reference/mibs/mib-jnx-if-extensions.txt.

For information about the enterprise-specific Interface MIB objects, see the following topics:

- [jnxIfTable on page 447](#)
- [ifChassisTable on page 450](#)

jnxIfTable

jnxIfTable lists traffic statistics, input errors, and output errors for an interface.

jnxIfEntry objects are listed in [Table 132 on page 447](#).

Table 132: jnxIfTable

Object	Object Identifier	Description
ifIn1SecRates	ifJnxEntry 1	The number of bits per second (bps) delivered by this sublayer to its next higher sublayer.
ifIn1SecOctets	ifJnxEntry 2	The number of octets per second delivered by this sublayer to its next higher sublayer.

Table 132: jnxIfTable (*continued*)

Object	Object Identifier	Description
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.
ifJnxInL3Incompletes	ifJnxEntry 17	L3 incompletes: The number of incoming packets that fail Layer 3 sanity checks of the header.

Table 132: jnxIfTable (*continued*)

Object	Object Identifier	Description
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.
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 number of CRC errors that have occurred.

Table 132: jnxIfTable (*continued*)

Object	Object Identifier	Description
ifJnxFcsErrors	ifJnxEntry 33	The number of FCS errors that have occurred.

ifChassisTable

ifChassisTable provides additional interface and chassis information.

ifChassisEntry objects are listed in [Table 133 on page 450](#).

Table 133: ifChassisTable

Object	Object Identifier	Description
ifChassisFpc	ifChassisEntry 1	<p>The number of the FPC card on which the interface is located in the chassis. It is the chassis slot in which the FPC card is installed for the specified interface.</p> <p>Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical FPC associated with the specified interface.</p>
ifChassisPic	ifChassisEntry 2	<p>The number of the PIC card on which the interface is located in the chassis. It is the PIC location on the FPC card for the specified interface.</p> <p>Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical PIC associated with the specified interface.</p>
ifChassisPort	ifChassisEntry 3	<p>The number of the port on the PIC card on which the interface is located in the chassis. It is the port number on the PIC card for the specified interface.</p> <p>Although the number is labeled from 0 and up in the chassis, the return value for this object always starts from 1 according to network management convention. Therefore, a value of zero means there is no real or physical port associated with the specified interface.</p>
ifChassisChannel	ifChassisEntry 4	<p>The channel identifier for the specified interface if it is part of a channelized interface.</p> <p>Although the channel is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to network management convention. For an interface that could not be channelized, this object returns zero.</p>

Table 133: ifChassisTable (*continued*)

Object	Object Identifier	Description
ifChassisLogicalUnit	ifChassisEntry 5	<p>The logical unit number of the specified interface. It is the logical part of the interface that is configured on the physical or channel part, if any.</p> <p>Although the logical unit number is numbered from 0 and up in the interface naming, the return value for this object always starts from 1 according to network management convention. For an interface that is really a physical device, this value returns zero.</p>
ifChassisPicIndex	ifChassisEntry 6	<p>The indexes for the Chassis MIB tables. This is the instance index that keys into jnxContentsTable in the Chassis MIB.</p> <p>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."</p>

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 34

IP Forward MIB

- [Interpreting the Enterprise-Specific IP Forward MIB on page 453](#)

Interpreting the Enterprise-Specific IP Forward MIB

- [IP Forwarding MIB on page 453](#)
- [jnxIpCidrRouteTable on page 454](#)
- [jnxInetCidrRouteTable on page 454](#)

IP Forwarding MIB

The Juniper Networks enterprise-specific IP Forwarding MIB, whose object identifier is **{jnxMibs 38}**, extends the **inetCidrRouteTable** in the IP Forwarding Table MIB (as defined in RFC 4292) to include a tunnel name when the next hop is through an RSVP-signaled LSP.

This MIB adds a **jnxInetCidrRouteTunnelName** attribute to the **inetCidrRouteTable**. The attribute exists for each entry in the **inetCidrRouteTable**. (One entry in the **inetCidrRouteTable** 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 **jnxInetCidrRouteTunnelName**. 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 **inetCidrRouteTable**, the index is identical to that used in the **inetCidrRouteTable**. 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/junos13.2/topics/reference/mibs/mib-jnx-ipforward.txt.

For more information about the enterprise-specific IP Forwarding MIB objects, see the following topic:

- [jnxIpCidrRouteTable](#) on page 454

jnxIpCidrRouteTable



NOTE: This has been replaced with “[jnxInetCidrRouteTable](#)” on page 454 in Junos OS Release 12.2.

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 134 on page 454](#).

Table 134: 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.

jnxInetCidrRouteTable

The **jnxInetCidrRouteTable**, whose object identifier is **{jnxInetForwardMIB 1}**, extends the **inetCidrRouteTable** with additional data.

jnxInetCidrRouteEntry, whose object identifier is **{jnxInetCidrRouteTable 1}**, has one object, which is listed in [Table 135 on page 454](#).

Table 135: jnxInetCidrRouteTable

Object	Object Identifier	Description
jnxInetCidrRouteTunnelName	jnxInetCidrRouteEntry 1	The canonical name assigned to the tunnel. The router forwards traffic bound for the destination through this tunnel.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

IPsec Generic Flow Monitoring Object MIB

- Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object MIB on page 457

Interpreting the Enterprise-Specific IPsec Generic Flow Monitoring Object MIB

- IPsec Generic Flow Monitoring Object MIB on page 457
- Branch Tree Objects on page 458
- IPsec Generic Flow Monitoring Object MIB Textual Conventions on page 458
- Number of IKE Tunnels Currently Active on page 462
- IPsec Phase 1 IKE Tunnel Table on page 462
- IPsec Phase 2 IKE Tunnel Table on page 465
- IPsec Phase 2 Security Association Table on page 469

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/junos13.2/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 458
- IPsec Generic Flow Monitoring Object MIB Textual Conventions on page 458
- Number of IKE Tunnels Currently Active on page 462

- [IPsec Phase 1 IKE Tunnel Table on page 462](#)
- [IPsec Phase 2 IKE Tunnel Table on page 465](#)
- [IPsec Phase 2 Security Association Table on page 469](#)

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`}

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 136 on page 458](#)).

Table 136: IKE Identity Type Textual Conventions

Type	Description	Syntax Integer
Unknown	Unknown type	(0)
<code>idIpv4Addr</code>	IPv4 address	(1)
<code>idFqdn</code>	Fully qualified domain name	(2)
<code>idDn</code>	Distinguished name	(3)
<code>idUfqdn</code>	User fully qualified domain name	(4)

- **JnxIkeNegoMode**—The IPsec Phase 1 IKE negotiation mode (see [Table 137 on page 458](#)).

Table 137: 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 138 on page 459](#)).

Table 138: IKE Negotiations Hash Algorithms

Hash Algorithm	Syntax Integer
md5	(1)
sha	(2)

- **JnxIkeAuthMethod**—The authentication method used in IPsec Phase 1 IKE negotiations (see [Table 139 on page 459](#)).

Table 139: IKE Authentication Method

Method	Syntax Integer
preSharedKey	(1)
dssSignature	(2)
rsaSignature	(3)
rsaEncryption	(4)
revRsaEncryption	(5)
xauthPreSharedKey	(6)
xauthDssSignature	(7)
xauthRsaSignature	(8)
xauthRsaEncryption	(9)
xauthRevRsaEncryption	(10)

- **JnxIkePeerRole**—The role of the local endpoint in negotiating the IPsec Phase 1 IKE security association (SA). It can be either initiator or responder (see [Table 140 on page 459](#)).

Table 140: Role of Local Endpoint in Negotiations

Role	Syntax Integer
Initiator	(1)
Responder	(2)

- **JnxIkeTunStateType**—State of the Phase 1 IKE negotiation (see [Table 141 on page 460](#)).

Table 141: State of Phase 1 IKE Negotiation

State	Syntax Integer
Up	(1)
Down	(2)

- **JnxDiffHellmanGrp**—The Diffie-Hellman Group used in negotiations (see [Table 142 on page 460](#)).

Table 142: 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 143 on page 460](#)).

Table 143: 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 144 on page 460](#)).

Table 144: Encryption Algorithm in Negotiations

Algorithm	Syntax Integer
espDes	(1)
esp3des	(2)
espNull	(3)
espAes128	(4)
espAes192	(5)

Table 144: Encryption Algorithm in Negotiations (*continued*)

Algorithm	Syntax Integer
espAes256	(6)

- **JnxAuthAlgo**—The authentication algorithm used by an SA of an IPsec Phase 2 Tunnel (see [Table 145 on page 461](#)).

Table 145: 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 146 on page 461](#)).

Table 146: 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 147 on page 461](#)).

Table 147: Role of Local Endpoint in Negotiations

SA Type	Syntax Integer
Unknown	(0)
Manual	(1)
Dynamic	(2)

Number of IKE Tunnels Currently Active



NOTE: The `jnxIkeNumOfTunnels` object is not supported in this release.

Table 148 on page 462 reports the number of IKE Tunnels currently active.

Table 148: Number of IKE Tunnels Currently Active

Object	Object ID	Description
<code>jnxIkeNumOfTunnels</code>	<code>jnxIpSecFlowMonPhaseOne</code> 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 <code>jnxIkeTunnelMonTable</code> .

IPsec Phase 1 IKE Tunnel Table

Table 149 on page 462 identifies objects listed in the IPsec Phase 1 IKE Tunnel Table.

Phase 1 is used to negotiate the parameter and key material required to establish an ISAKMP SA.

Phase 1 SA components include an encryption algorithm, authentication, Diffie-Hellman group values, and authentication methods, such as preshared keys or certificates.

Table 149: IPsec Phase 1 IKE Tunnel Table

Object	Object ID	Description
<code>jnxIkeTunnelMonTable</code>	<code>jnxIpSecFlowMonPhaseOne</code> 2	The IPsec Phase 1 IKE Tunnel Table. There is one entry in this table for each active IPsec Phase 1 IKE tunnel.

Table 149: IPsec Phase 1 IKE Tunnel Table (*continued*)

Object	Object ID	Description
<code>jnxIkeTunnelMonEntry</code>	<code>jnxIkeTunnelMonTable 1</code>	<p>Attributes associated with an active IPsec Phase 1 IKE tunnel.</p> <p>Sequence of attributes:</p> <ul style="list-style-type: none"> • <code>jnxIkeTunMonRemoteGwAddrType</code> • <code>jnxIkeTunMonRemoteGwAddr</code> • <code>jnxIkeTunMonIndex</code> • <code>jnxIkeTunMonLocalGwAddrType</code> • <code>jnxIkeTunMonLocalGwAddr</code> • <code>jnxIkeTunMonState</code> • <code>jnxIkeTunMonInitiatorCookie</code> • <code>jnxIkeTunMonResponderCookie</code> • <code>jnxIkeTunMonLocalRole</code> • <code>jnxIkeTunMonLocalIdType</code> • <code>jnxIkeTunMonLocalIdValue</code> • <code>jnxIkeTunMonLocalCertName</code> • <code>jnxIkeTunMonRemoteldType</code> • <code>jnxIkeTunMonRemoteldValue</code> • <code>jnxIkeTunMonNegoMode</code> • <code>jnxIkeTunMonDiffHellmanGrp</code> (not supported in this release) • <code>jnxIkeTunMonEncryptAlgo</code> • <code>jnxIkeTunMonHashAlgo</code> • <code>jnxIkeTunMonAuthMethod</code> • <code>jnxIkeTunMonLifeTime</code> • <code>jnxIkeTunMonActiveTime</code> • <code>jnxIkeTunMonInOctets</code> • <code>jnxIkeTunMonInPkts</code> • <code>jnxIkeTunMonOutOctets</code> • <code>jnxIkeTunMonOutPkts</code> • <code>jnxIkeTunMonXAuthUserId</code> • <code>jnxIkeTunMonDPDDownCount</code>
<code>jnxIkeTunMonRemoteGwAddrType</code>	<code>jnxIkeTunnelMonEntry 1</code>	IP address type of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel.
<code>jnxJsFwAuthClientIpAddr</code>	<code>jnxJsAuthTrapVars 4</code>	IP address of remote gateway (endpoint) for the IPsec Phase 1 IKE tunnel.
<code>jnxIkeTunMonIndex</code>	<code>jnxIkeTunnelMonEntry 3</code>	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.
<code>jnxIkeTunMonLocalGwAddr</code>	<code>jnxIkeTunnelMonEntry 4</code>	IP address of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel.

Table 149: IPsec Phase 1 IKE Tunnel Table (*continued*)

Object	Object ID	Description
<code>jnxIkeTunMonLocalGwAddrType</code>	<code>jnxIkeTunnelMonEntry 5</code>	IP address type of local endpoint (gateway) for the IPsec Phase 1 IKE tunnel.
<code>jnxIkeTunMonState</code>	<code>jnxIkeTunnelMonEntry 6</code>	State of IKE tunnel. It can be: <ul style="list-style-type: none"> • 1—up, negotiation completed. • 2—down, being negotiated.
<code>jnxIkeTunMonInitiatorCookie</code>	<code>jnxIkeTunnelMonEntry 7</code>	Cookie as generated by peer that initiated the IKE Phase 1 negotiation. This cookie is carried in the ISAKMP header.
<code>jnxIkeTunMonResponderCookie</code>	<code>jnxIkeTunnelMonEntry 8</code>	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.
<code>jnxIkeTunMonLocalRole</code>	<code>jnxIkeTunnelMonEntry 9</code>	Role of local peer identity. The role of the local peer can be: <ul style="list-style-type: none"> • Initiator • Responder
<code>jnxIkeTunMonLocalIdType</code>	<code>jnxIkeTunnelMonEntry 10</code>	Type of local peer identity. The local peer can be identified by: <ul style="list-style-type: none"> • IP address • Fully qualified domain name string • Distinguished name string
<code>jnxIkeTunMonLocalIdValue</code>	<code>jnxIkeTunnelMonEntry 11</code>	Value of local peer identity. <p>If the local peer type is an IP address, then this is the IP address used to identify the local peer.</p> <p>If the local peer type is a fully qualified domain name string, then this is the fully qualified domain name string of the local peer.</p> <p>If the local peer type is a distinguished name string, then this is the distinguished name string of the local peer.</p>
<code>jnxIkeTunMonLocalCertName</code>	<code>jnxIkeTunnelMonEntry 12</code>	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.
<code>jnxIkeTunMonRemoteldType</code>	<code>jnxIkeTunnelMonEntry 13</code>	Type of remote peer identity. The remote peer can be identified by: <ul style="list-style-type: none"> • IP address • Fully qualified domain name string • Distinguished name string

Table 149: IPsec Phase 1 IKE Tunnel Table (*continued*)

Object	Object ID	Description
<code>jnxIkeTunMonRemoteldValue</code>	<code>jnxIkeTunnelMonEntry 14</code>	<p>Value of remote peer identity.</p> <p>If the remote peer type is an IP address, then this is the IP address used to identify the remote peer.</p> <p>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.</p> <p>If the remote peer type is a distinguished name string, then this is the distinguished name string of the remote peer.</p>
<code>jnxIkeTunMonNegoMode</code>	<code>jnxIkeTunnelMonEntry 15</code>	Negotiation mode of IPsec Phase 1 IKE tunnel.
NOTE: The <code>jnxIkeTunMonDiffHellmanGrp</code> object is not supported in this release.		
<code>jnxIkeTunMonDiffHellmanGrp</code>	<code>jnxIkeTunnelMonEntry 16</code>	Diffie-Hellman Group used in IPsec Phase 1 IKE negotiations.
<code>jnxIkeTunMonEncryptAlgo</code>	<code>jnxIkeTunnelMonEntry 17</code>	Encryption algorithm used in IPsec Phase 1 IKE negotiations.
<code>jnxIkeTunMonHashAlgo</code>	<code>jnxIkeTunnelMonEntry 18</code>	Hash algorithm used in IPsec Phase 1 IKE negotiations.
<code>jnxIkeTunMonAuthMethod</code>	<code>jnxIkeTunnelMonEntry 19</code>	Authentication method used in IPsec Phase 1 IKE negotiations.
<code>jnxIkeTunMonLifeTime</code>	<code>jnxIkeTunnelMonEntry 20</code>	Negotiated lifetime of IPsec Phase 1 IKE tunnel in seconds.
<code>jnxIkeTunMonActiveTime</code>	<code>jnxIkeTunnelMonEntry 21</code>	Length of time IPsec Phase 1 IKE tunnel has been active in hundredths of seconds.
<code>jnxIkeTunMonInOctets</code>	<code>jnxIkeTunnelMonEntry 22</code>	Total number of octets received by this IPsec Phase 1 IKE SA.
<code>jnxIkeTunMonInPkts</code>	<code>jnxIkeTunnelMonEntry 23</code>	Total number of packets received by this IPsec Phase 1 IKE SA.
<code>jnxIkeTunMonOutOctets</code>	<code>jnxIkeTunnelMonEntry 24</code>	Total number of octets sent by this IPsec Phase 1 IKE SA.
<code>jnxIkeTunMonOutPkts</code>	<code>jnxIkeTunnelMonEntry 25</code>	Total number of packets sent by this IPsec Phase 1 IKE SA.
<code>jnxIkeTunMonXAuthUserId</code>	<code>jnxIkeTunnelMonEntry 26</code>	Extended Authentication (XAuth) User Identifier. Identifies the user associated with this IPsec Phase 1 negotiation.
<code>jnxIkeTunMonDPDDownCount</code>	<code>jnxIkeTunnelMonEntry 27</code>	Number of times that the remote peer is detected in a dead (or down) state.

IPsec Phase 2 IKE Tunnel Table

Table 150 on page 466 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 150: IPsec Phase 2 IKE Tunnel Table

Object	Object ID	Description
--------	-----------	-------------

NOTE: The `jnxIpSecNumOfTunnels` object is not supported in this release.

<code>jnxIpSecNumOfTunnels</code>	<code>jnxIpSecFlowMonPhaseTwo</code> 1	Number of IPsec VPN tunnels. This attribute should report the number of IPsec VPN tunnels in <code>jnxIpSecTunnelTable</code> .
<code>jnxIpSecTunnelMonTable</code>	<code>jnxIpSecFlowMonPhaseTwo</code> 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.

Table 150: IPsec Phase 2 IKE Tunnel Table (*continued*)

Object	Object ID	Description
<code>jnxIpSecTunnelMonEntry</code>	<code>jnxIpSecTunnelMonTable 1</code>	<p>Each entry contains the attributes associated with an active IPsec Phase 2 tunnel.</p> <p>Sequence of attributes:</p> <ul style="list-style-type: none"> • <code>jnxIpSecTunMonRemoteGwAddrType</code> • <code>jnxIpSecTunMonRemoteGwAddr</code> • <code>jnxIpSecTunMonIndex</code> • <code>jnxIpSecTunMonLocalGwAddrType</code> • <code>jnxIpSecTunMonLocalGwAddr</code> • <code>jnxIpSecTunMonLocalProxyId</code> • <code>jnxIpSecTunMonRemoteProxyId</code> • <code>jnxIpSecTunMonKeyType</code> • <code>jnxIpSecTunMonRemotePeerType</code> • <code>jnxIpSecTunMonOutEncryptedBytes</code> • <code>jnxIpSecTunMonOutEncryptedPkts</code> • <code>jnxIpSecTunMonInDecryptedBytes</code> • <code>jnxIpSecTunMonInDecryptedPkts</code> • <code>jnxIpSecTunMonAHInBytes</code> • <code>jnxIpSecTunMonAHInPkts</code> • <code>jnxIpSecTunMonAHOOutBytes</code> • <code>jnxIpSecTunMonAHOOutPkts</code> • <code>jnxIpSecTunMonReplayDropPkts</code> • <code>jnxIpSecTunMonAhAuthFails</code> • <code>jnxIpSecTunMonDecryptFails</code> • <code>jnxIpSecTunMonBadHeaders</code> • <code>jnxIpSecTunMonBadTrailers</code> • <code>jnxIpSecTunMonOutOctets</code> • <code>jnxIpSecTunMonDroppedPkts</code> (not supported in this release)
<code>jnxIpSecTunMonRemoteGwAddrType</code>	<code>jnxIpSecTunnelMonEntry 1</code>	IP address type of remote gateway (endpoint) for the IPsec Phase 2 tunnel.
<code>jnxIpSecTunMonRemoteGwAddr</code>	<code>jnxIpSecTunnelMonEntry 2</code>	IP address of remote gateway (endpoint) for the IPsec Phase 2 tunnel.
<code>jnxIpSecTunMonIndex</code>	<code>jnxIpSecTunnelMonEntry 3</code>	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.
<code>jnxIpSecTunMonLocalGwAddrType</code>	<code>jnxIpSecTunnelMonEntry 4</code>	IP address type of local gateway (endpoint) for the IPsec Phase 2 tunnel.
<code>jnxIpSecTunMonLocalGwAddr</code>	<code>jnxIpSecTunnelMonEntry 5</code>	IP address of local gateway (endpoint) for the IPsec Phase 2 tunnel.

Table 150: 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: <ul style="list-style-type: none"> • IKE-negotiated • Manually installed
jnxIpSecTunMonRemotePeerType	jnxIpSecTunnelMonEntry 9	Type of the remote peer gateway (endpoint). It can be one of the following two types: <ul style="list-style-type: none"> • 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.

Table 150: IPsec Phase 2 IKE Tunnel Table (*continued*)

Object	Object ID	Description
<code>jnxIpSecTunMonBadHeaders</code>	<code>jnxIpSecTunnelMonEntry 22</code>	Number of packets received by this Phase 2 tunnel that failed due to bad headers.
<code>jnxIpSecTunMonBadTrailers</code>	<code>jnxIpSecTunnelMonEntry 23</code>	Number of packets received by this Phase 2 tunnel that failed due to bad ESP trailers.
NOTE: The <code>jnxIpSecTunMonDroppedPkts</code> object is not supported in this release.		
<code>jnxIpSecTunMonDroppedPkts</code>	<code>jnxIpSecTunnelMonEntry 26</code>	Total number of dropped packets for this Phase 2 tunnel.

IPsec Phase 2 Security Association Table

`jnxIpSecSaMonTable`, whose object ID is `{jnxIpSecFlowMonPhaseTwo 3}`, identifies the objects listed in [Table 151 on page 470](#). 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 151: IPsec Phase 2 Security Association Table

Object	Object ID	Description
jnxIpSecSaMonEntry	jnxIpSecSaMonTable 1	<p>Each entry contains the attributes associated with active and expiring IPsec Phase 2 SAs.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • 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.

Table 151: 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**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

IPsec Monitoring MIB

- [Interpreting the Enterprise-Specific IPsec Monitoring MIB on page 473](#)

Interpreting the Enterprise-Specific IPsec Monitoring MIB

- [IPSec Monitoring MIB on page 473](#)
- [jnxIkeTunnelTable on page 473](#)
- [jnxIPSecTunnelTable on page 476](#)
- [jnxIPSecSaTable on page 478](#)

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/junos13.2/topics/reference/mibs/mib-jnx-ipsec-monitor-asp.txt.

For information about IPsec Monitoring MIB objects, see the following topics:

- [jnxIkeTunnelTable on page 473](#)
- [jnxIPSecTunnelTable on page 476](#)
- [jnxIPSecSaTable on page 478](#)

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 152 on page 474](#). Each entry contains attributes associated with an active IPsec phase 1 IKE tunnel.

Table 152: jnxIkeTunnelTable

Object	Object Identifier	Description
<code>jnxIkeTunIndex</code>	<code>jnxIkeTunnelEntry 1</code>	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.
<code>jnxIkeTunLocalRole</code>	<code>jnxIkeTunnelEntry 2</code>	The role of the local peer identity. The role can be initiator or responder .
<code>jnxIkeTunNegState</code>	<code>jnxIkeTunnelEntry 3</code>	The state of the current negotiation. The state can be matured or non matured .
<code>jnxIkeTunInitiatorCookie</code>	<code>jnxIkeTunnelEntry 4</code>	Cookie generated by the peer that initiated the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
<code>jnxIkeTunResponderCookie</code>	<code>jnxIkeTunnelEntry 5</code>	Cookie generated by the peer responding to the IKE phase 1 negotiation. This cookie is carried in the ISAKMP header.
<code>jnxIkeTunLocalIdType</code>	<code>jnxIkeTunnelEntry 6</code>	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.
<code>jnxIkeTunLocalIdValue</code>	<code>jnxIkeTunnelEntry 7</code>	The value of the local peer identity. <ul style="list-style-type: none"> • 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(<code>if_fqdn</code>), then this is the FQDN of the remote peer. • If the local peer type is a distinguished name (<code>id_dn</code>), then this is the distinguished name of the local peer.
<code>jnxIkeTunLocalGwAddrType</code>	<code>jnxIkeTunnelEntry 8</code>	The IP address type of the local endpoint (gateway) for the IPsec phase 1 IKE tunnel.
<code>jnxIkeTunLocalGwAddr</code>	<code>jnxIkeTunnelEntry 9</code>	The IP address of the local endpoint (gateway) for the IPsec phase 1 IKE tunnel.

Table 152: jnxIkeTunnelTable (*continued*)

Object	Object Identifier	Description
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. <ul style="list-style-type: none"> • 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.
jnxIkeTunActiveTime	jnxIkeTunnelEntry 21	The length of time (in hundredths of seconds) that the IPsec phase 1 IKE tunnel has been active.

Table 152: jnxIkeTunnelTable (continued)

Object	Object Identifier	Description
jnxIkeTunInOctets	jnxIkeTunnelEntry 22	The total number of octets received by this IPsec phase 1 IKE security association.
jnxIkeTunInPkts	jnxIkeTunnelEntry 23	The total number of packets received by this IPsec phase 1 IKE security association.
jnxIkeTunOutOctets	jnxIkeTunnelEntry 24	The total number of octets sent by this IPsec phase 1 IKE security association.
jnxIkeTunOutPkts	jnxIkeTunnelEntry 25	The total number of octets sent by this IPsec phase 1 IKE security association.

jnxIPSecTunnelTable

The IPsec phase 2 tunnel table (**jnxIPSecTunnelTable**), whose object identifier is **{jnxIPSecPhaseTwo 1}**, is used to monitor the IPsec phase 2 tunnel attributes along with the statistics 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 153 on page 476](#). Each entry contains attributes associated with an active IPsec phase 2 tunnel.

Table 153: 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.

Table 153: jnxIPSecTunnelTable (*continued*)

Object	Object Identifier	Description
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.
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.

Table 153: jnxIPSecTunnelTable (*continued*)

Object	Object Identifier	Description
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.

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 154 on page 479](#). Each entry contains SA components for an active IPsec phase 2 tunnel.

Table 154: 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.
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).

Table 154: jnxIPSecSaTable (*continued*)

Object	Object Identifier	Description
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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 37

IPsec VPN Objects MIB

- [Interpreting the Enterprise-Specific IPsec VPN Objects MIB on page 481](#)

Interpreting the Enterprise-Specific IPsec VPN Objects MIB

- [IPsec VPN Objects MIB on page 481](#)
- [IPSec VPN Objects MIB Textual Conventions on page 482](#)
- [jnxJsIpSecTunnelTable on page 482](#)

IPsec VPN Objects MIB

The Juniper Networks enterprise-specific IPsec VPN Objects MIB, **jnxJsIpSecVpnMib**, whose object ID is **{jnxJsIPSecVpn 1}**, defines the object used to monitor the entries pertaining to IPsec objects and the management of the IPsec VPN functionalities for Juniper Networks security product lines. This MIB models IPsec attributes specific to the appropriate Juniper Networks implementation.

This MIB module extends the Juniper Networks common IPsec flow monitoring MIB. Building on the existing common infrastructure, the security implementation integrates the value-added features for the security products.

Related IPsec VPN Objects MIBs include:

- **jnxJsIpSecVpnNotifications** {jnxJsIpSecVpnMib 0}
- **jnxJsIpSecVpnPhaseOne** {jnxJsIpSecVpnMib 1}
- **jnxJsIpSecVpnPhaseTwo** {jnxJsIpSecVpnMib 2}

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-js-ipsec-vpn.txt.

For information about the enterprise-specific IPsec VPN Objects MIB, see the following topics:

- [IPSec VPN Objects MIB Textual Conventions on page 482](#)
- [jnxJsIpSecTunnelTable on page 482](#)

IPSec VPN Objects MIB Textual Conventions

Table 155 on page 482 explains the possible values for **jnxJsIpSecVpnType**, which is the type of remote peer gateway (endpoint).

Table 155: jnxJsIpSecVpnType

Type	Description	Syntax Integer
policyBased	Tunnels require a policy with action tunnel to trigger IPsec VPN. The device receives traffic and matches it with a policy that has action tunnel , then performs the encryption/decryption and authentication options negotiated for this VPN Phase 2 negotiation.	(1)
routeBased	Requires a tunnel interface to a route directing traffic to protected networks to exit the system using that tunnel interface. The tunnel interface is bound to a Phase 2 VPN configuration that specifies all the tunnel parameters.	(2)

jnxJsIpSecTunnelTable

jnxJsIpSecTunnelTable, whose object ID is {**jnxJsIpSecVpnPhaseTwo1**}, is the IPsec Phase 2 Tunnel Table, with objects listed in Table 156 on page 482. There is one entry for each active IPsec Phase 2 tunnel. If the tunnel is terminated, then the entry is no longer available after the table has been refreshed.

This table augments **jnxIpSecTunnelMonTable** in “IPsec Phase 2 IKE Tunnel Table” on page 465.

Table 156: jnxJsIpSecTunnelTable

Object	Object ID	Description
jnxJsIpSecTunnelEntry	jnxJsIpSecTunnelTable 1	Each entry contains the attributes associated with an active IPsec Phase 2 tunnel. Sequence of parameters: <ul style="list-style-type: none"> • jnxJsIpSecTunPolicyName • jnxJsIpSecVpnTunType • jnxJsIpSecTunCfgMonState • jnxJsIpSecTunState
jnxJsIpSecTunPolicyName	jnxJsIpSecTunnelEntry 1	Policy name associated with this tunnel if the IPsec VPN is policy-based. If the IPsec VPN is not policy-based, this attribute is not applicable.

Table 156: jnxJslpSecTunnelTable (*continued*)

Object	Object ID	Description
jnxJslpSecVpnTunType	jnxJslpSecTunnelEntry 2	Attribute to indicate whether the IPsec VPN tunnel is policy-based or route-based.
jnxJslpSecTunCfgMonState	jnxJslpSecTunnelEntry 3	<p>According to user configuration, whether to monitor the IPsec tunnel to be enabled or not:</p> <ul style="list-style-type: none"> • disable—(1) • enable—(2)
jnxJslpSecTunState	jnxJslpSecTunnelEntry 4	<p>Attribute to indicate whether the IPsec tunnel is up or down, determined by ICMP ping if jnxJslpSecTunCfgMonState is enabled:</p> <ul style="list-style-type: none"> • up—(1): VPN monitor detects the tunnel is up. • down—(2): VPN monitor detects the tunnel is down. • vpnMonitoringDisabled—(3): User has disabled VPN tunnel monitoring.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 38

IPv4 MIB

- Interpreting the Enterprise-Specific IPv4 MIB on page 485
- Standard IPv4/IPv6 MIBs on page 486

Interpreting the Enterprise-Specific IPv4 MIB

- IPv4 MIB on page 485
- jnxIpv4AddrTable on page 485

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/junos13.2/topics/reference/mibs/mib-jnx-ipv4.txt.

For information about the enterprise-specific IPv4 MIB objects, see the following topic:

- jnxIpv4AddrTable on page 485

jnxIpv4AddrTable

The **jnxIpv4AddrTable** defines the **jnxIpv4AddrEntry** and its attributes. Each **jnxIpv4AddrEntry** contains the objects listed in [Table 157 on page 485](#).

Table 157: 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 157: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Standard IPv4/IPv6 MIBs

The MIBs listed in this section are required to bring the system-wide statistics support in RFC 4293 for both IPv4 and IPv6 versions in Juniper Networks routers. With this support, the Internet service providers (ISPs) can manage IPv4 or IPv6 system-wide traffic statistics in a consistent manner. This feature keeps account of IPv6-specific packets and bytes traversing through the router. The feature can be used by ISPs to track the amount of IPv6 traffic going through their network compared to IPv4 traffic.

See [Table 158 on page 487](#) for a list of standard MIB groups implemented so that the ISPs can manage IPv4 or IPv6 system-wide traffic statistics in a consistent manner. See [Table 159 on page 502](#) for list of MIB objects supported by MX Series and EX Series devices. See [Table 160 on page 504](#) for list of supported Internet Control Message Protocol version 4 (ICMPv4) types and [Table 161 on page 505](#) for a list of supported ICMPv6 types by MX Series and EX Series devices.

For RFC information, go to: <http://www.ietf.org/rfc/rfc4293.txt>.

Table 158: Standard MIB Groups for IPv6

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipSystemStatsGroup	IP system wide statistics.	ipSystemStatsInReceives	The total number of input IP datagrams received, including those received in error. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInOctets	The total number of octets received in input IP datagrams, including those received in error. Octets from datagrams counted in ipSystemStatsInReceives <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime	{ipMIBGroups 8}
		ipSystemStatsInHdrErrors	The number of input IP datagrams discarded due to errors in their IP headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IP options, and so on. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInNoRoutes	The number of input IP datagrams discarded because no route could be found to transmit them to their destination. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInAddrErrors		{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The number of input IP datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, ::0). For entities that are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsInUnknownProtos	The number of locally addressed IP datagrams received successfully but discarded because of an unknown or unsupported protocol. When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInTruncatedPkts	The number of input IP datagrams discarded because the datagram frame did not carry enough data. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInFowDatagrams		{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful. When tracking interface statistics, the counter of the incoming interface is incremented for each datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsReasmReqds	The number of IP fragments received that needed to be reassembled at this interface. When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsReasmOKs	The number of IP datagrams successfully reassembled. When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsReasmFails		{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, and so on).</p> <p>NOTE: This is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received. When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	
		<code>ipSystemStatsInDiscards</code>	<p>The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but were discarded (for example, for lack of buffer space).</p> <p>NOTE: This counter does not include any datagrams discarded while awaiting re-assembly.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>
		<code>ipSystemStatsInDelivers</code>		<code>{ipMIBGroups 8}</code>

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of datagrams successfully delivered to IP user protocols (including ICMP). When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsOutRequests	<p>The total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission.</p> <p>NOTE: This counter does not include any datagrams counted in ipSystemStatsOutForwDatagrams.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.</p>	{ipMIBGroups 8}
		ipSystemStatsOutNoRoutes	<p>The number of locally generated IP datagrams discarded because no route could be found to transmit them to their destination.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.</p>	{ipMIBGroups 8}
		ipSystemStatsOutForwDatagrams		{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully forwarded datagram.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	
		<code>ipSystemStatsOutDiscards</code>	<p>The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but were discarded (for example, for lack of buffer space).</p> <p>NOTE: This counter would include datagrams counted in <code>ipSystemStatsOutForwDatagrams</code> if any such datagrams met this (discretionary) discard criterion.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>
		<code>ipSystemStatsOutFragReqds</code>	<p>The number of IP datagrams that would require fragmentation in order to be transmitted. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
		ipSystemStatsOutFragOKs	The number of IP datagrams that have been successfully fragmented. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutFragFails	The number of IP datagrams that have been discarded because they needed to be fragmented but could not be. This includes IPv4 packets that have the DF bit set and IPv6 packets that are being forwarded and exceed the outgoing link MTU. When tracking interface statistics, the counter of the outgoing interface is incremented for an unsuccessfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutFragCreates	The number of output datagram fragments that have been generated as a result of IP fragmentation. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutTransmits		{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of IP datagrams that this entity supplied to the lower layers for transmission. This includes datagrams generated locally and those forwarded by this entity. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsOutOctets	The total number of octets in IP datagrams delivered to the lower layers for transmission. Octets from datagrams counted in ipSystemStatsOutTransmits <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInMcastPkts	The number of IP multicast datagrams received. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInMcastOctets	The total number of octets received in IP multicast datagrams. Octets from datagrams counted in ipSystemStatsInMcastPkts <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutMcastPkts	The number of IP multicast datagrams transmitted. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutMcastOctets		

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of octets transmitted in IP multicast datagrams. Octets from datagrams counted in ipSystemStatsOutMcastPkts <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsDiscontinuityTime	The value of sysUpTime on the most recent occasion at which any one or more of this entry's counters suffered a discontinuity. If no such discontinuities have occurred since the last re- initialization of the local management subsystem, then this object contains a zero value.	{ipMIBGroups 8}
		ipSystemStatsRefreshRate	The minimum reasonable polling interval for this entry. This object provides an indication of the minimum amount of time required to update the counters in this entry.	{ipMIBGroups 8}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipAddressGroup	The group of objects for providing information about the addresses relevant to this entity's interfaces.	ipAddressSpinLock	<p>An advisory lock used to allow cooperating SNMP managers to coordinate their use of the set operation in creating or modifying rows within this table. In order to use this lock to coordinate the use of set operations, managers should first retrieve ipAddressTableSpinLock. They should then determine the appropriate row to create or modify. Finally, they should issue the appropriate set command, including the retrieved value of ipAddressSpinLock. If another manager has altered the table in the meantime, then the value of ipAddressSpinLock will have changed, and the creation will fail as it will be specifying an incorrect value for ipAddressSpinLock. It is suggested, but not required, that the ipAddressSpinLock be the first var bind for each set of objects representing a 'row' in a PDU.</p> <p>An agent is not required to provide write access to this object. However, if an agent provides write access to any of the other objects in the ipAddressGroup, it <i>should</i> provide write access to this object as well.</p>	{ipMIBGroups 19}
		ipAddressIfIndex	<p>The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressType	<p>The type of address.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressStatus		{ipMIBGroups 19}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The status of the address, describing if the address can be used for communication. In the absence of other information, an IPv4 address is always preferred.</p> <p>An agent is not required to provide write or create access to this object.</p>	
		ipAddressRowStatus	<p>The status of this conceptual row. The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified. A conceptual row cannot be made active until the ipAddressIfIndex has been set to a valid index.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressStorageType	<p>The storage type for this conceptual row. If this object has a value of 'permanent', then no other objects are required to be able to be modified.</p> <p>An agent is not required to provide write or create access to this object. If an agent allows this object to be written or created, it is not required to allow this object to be set to readOnly, permanent, or nonVolatile.</p>	{ipMIBGroups 19}
		ipAddressPrefix	<p>A pointer to the row in the prefix table to which this address belongs. Can be { 0 0 } if there is no such row.</p>	{ipMIBGroups 19}
		ipAddressOrigin	<p>The origin of the address.</p>	{ipMIBGroups 19}
		ipAddressCreated	<p>The value of sysUpTime at the time this entry was created. If this entry was created prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.</p>	{ipMIBGroups 19}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
		ipAddressLastChanged	The value of sysUpTime at the time this entry was last updated. If this entry was updated prior to the last re- initialization of the local network management subsystem, then this object contains a zero value.	{ipMIBGroups 19}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipNetToPhysicalGroup	The group of objects for providing information about the mappings of the network address to the physical address known to this node.	ipNetToPhysicalPhysAddress	The media-dependent 'physical' address. As the entries in this table are typically not persistent when this object is written, the entity <i>should not</i> save the change to non volatile storage. An agent is not required to provide write or create access to this object.	{ipMIBGroups 20}
		ipNetToPhysicalLastUpdated	The value of sysUpTime at the time this entry was last updated. If this entry was updated prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.	{ipMIBGroups 20}
		ipNetToPhysicalType		{ipMIBGroups 20}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToPhysicalTable. That is, it effectively disassociates the interface identified with that entry from the mapping identified with that entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToPhysicalType object. The 'dynamic(3)' type indicates that the IP address to physical addresses mapping has been dynamically resolved using for example, IPv4 ARP or the IPv6 Neighbor Discovery protocol. The 'static(4)' type indicates that the mapping has been statically configured. Both of these types refer to entries that provide mappings for other entities' addresses. The 'local(5)' type indicates that the mapping is provided for an entity's own interface address. As the <i>entries</i> in this table are typically not persistent when this object is written, the entity <i>should not</i> save the change to nonvolatile storage.</p> <p>An agent is not required to provide write or create access to this object.</p>	
		ipNetToPhysicalState	The Neighbor Unreachability Detection state for the interface when the address mapping in this entry is used.	{ipMIBGroups 20}
		ipNetToPhysicalRowStatus		{ipMIBGroups 20}

Table 158: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The status of this conceptual row. The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified. A conceptual row cannot be made active until the ipNetToPhysicalPhysAddress object has been set. Note that if the ipNetToPhysicalType is set to 'invalid', the managed node might delete the entry independent of the state of this object.	
icmpStatsGroup	The group of objects providing ICMP statistics.	icmpStatsInMsgs	The total number of ICMP messages that the entity received. Note that this counter includes all those counted by icmpStatsInErrors.	{ipMIBGroups 24}
		icmpStatsInErrors	The number of ICMP messages that the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, and so on).	{ipMIBGroups 24}
		icmpStatsOutMsgs	The total number of ICMP messages that the entity attempted to send. Note that this counter includes all those counted by icmpStatsOutErrors.	{ipMIBGroups 24}
		icmpStatsOutErrors	The number of ICMP messages that this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value should not include errors discovered outside the ICMP layer, such as the inability of IP to route the resultant datagram. In some implementations, there might be no types of error that contribute to this counter's value.	{ipMIBGroups 24}
		icmpMsgStatsInPkts	The number of input packets for this AF and type.	{ipMIBGroups 24}
		icmpMsgStatsOutPkts	The number of output packets for this AF and type.	{ipMIBGroups 24}

Table 159: MIB Objects Supported by MX Series and EX Series

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
ipSystemStatsInReceives	Yes	Yes
ipSystemStatsInOctets	Yes	No
ipSystemStatsInHdrErrors	Yes	No
ipSystemStatsInNoRoutes	Yes	Yes
ipSystemStatsInAddrErrors	Yes	No
ipSystemStatsInUnknownProtos	Yes	Yes
ipSystemStatsInTruncatedPkts	Yes	No
ipSystemStatsInForwDatagrams	Yes	Yes
ipSystemStatsReasmReqds	Yes	Yes
ipSystemStatsReasmOKs	Yes	Yes
ipSystemStatsReasmFails	Yes	Yes
ipSystemStatsInDiscards	Yes	No
ipSystemStatsInDelivers	Yes	Yes
ipSystemStatsOutRequests	Yes	Yes
ipSystemStatsOutNoRoutes	Yes	Yes
ipSystemStatsOutForwDatagrams	Yes	Yes
ipSystemStatsOutDiscards	Yes	No
ipSystemStatsOutFragReqds	Yes	No
ipSystemStatsOutFragOKs	Yes	Yes
ipSystemStatsOutFragFails	Yes	Yes
ipSystemStatsOutFragCreates	Yes	Yes
ipSystemStatsOutTransmits	Yes	Yes
ipSystemStatsOutOctets	Yes	No
ipSystemStatsInMcastPkts	Yes	No

Table 159: MIB Objects Supported by MX Series and EX Series (*continued*)

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
ipSystemStatsInMcastOctets	Yes	No
ipSystemStatsOutMcastPkts	Yes	No
ipSystemStatsOutMcastOctets	Yes	No
ipSystemStatsDiscontinuityTime	Yes	No
ipSystemStatsRefreshRate	Yes	No
ipAddressSpinLock	Yes	No
ipAddressIfIndex	Yes	No
ipAddressType	Yes	No
ipAddressOrigin	Yes	No
ipAddressStatus	Yes	No
ipAddressCreated	Yes	Yes
ipAddressLastChanged	Yes	Yes
ipAddressRowStatus	Yes	Yes
ipAddressStorageType	Yes	No
ipNetToPhysicalPhysAddress	Yes	No
ipNetToPhysicalLastUpdated	Yes	Yes
ipNetToPhysicalType	Yes	No
ipNetToPhysicalState	Yes	Yes
ipNetToPhysicalRowStatus	Yes	No
icmpStatsInMsgs	Yes	Yes
icmpStatsInErrors	Yes	Yes
icmpStatsOutMsgs	Yes	Yes
icmpStatsOutErrors	Yes	Yes
icmpMsgStatsInPkts	Yes	Yes

Table 159: MIB Objects Supported by MX Series and EX Series (*continued*)

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
icmpMsgStatsOutPkts	Yes	Yes

Table 160: Supported ICMPv4 Counters

Counter Type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
0	Echo Reply	Yes	Yes
3	Destination Unreachable	Yes	Yes
4	Source Quench (Deprecated)	Yes	No
5	Redirect	Yes	Yes
6	Alternate Host Address	Yes	No
8	Echo	Yes	Yes
9	Router Advertisement	Yes	No
10	Router Selection	Yes	No
11	Time Exceeded	Yes	Yes
12	Parameter Problem	Yes	Yes
13	Timestamp	Yes	No
14	Timestamp Reply	Yes	Yes
15	Information Request	Yes	No
16	Information Reply	Yes	No
17	Address Mask Request	Yes	No
18	Address Mask Reply	Yes	No
30	Traceroute	Yes	No
31	Datagram Conversion Error	Yes	No
32	Mobile Host Redirect	Yes	—
33	IPv6 Where-Are-You	Yes	—
34	IPv6 I-Am-Here	Yes	—

Table 160: Supported ICMPv4 Counters (*continued*)

Counter Type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
35	Mobile Registration Request	Yes	–
36	Mobile Registration Reply	Yes	–
39	SKIP	Yes	–
40	Photuris	Yes	–

Table 161: Supported ICMPv6 Counters

Counter type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
1	Destination Unreachable	Yes	Yes
2	Packet Too Big	Yes	No
3	Time Exceeded	Yes	No
4	Parameter Problem	Yes	Yes
128	Echo Request	Yes	Yes
129	Echo Reply	Yes	Yes
130	Multicast Listener Query	Yes	Yes
131	Multicast Listener Report	Yes	Yes
132	Multicast Listener Done	Yes	Yes
133	Router Solicitation	Yes	–
134	Router Advertisement	Yes	Yes
135	Neighbor Solicitation	Yes	Yes
136	Neighbor Advertisement	Yes	Yes
137	Redirect Message	Yes	No
138	Router Renumbering	Yes	–
139	ICMP Node Information Query	Yes	–
140	ICMP Node Information Response	Yes	–
143	V2 Multicast Listener Report	Yes	–



NOTE: The conventions (Yes, No, –) indicate the following information:

- Yes: Supported (Junos OS supports statistics for the packet type)
 - No: Not supported (Junos OS does not support statistics for the packet type)
 - – : Not applicable (Junos OS does not support the packet type)
-

**Related
Documentation**

- *Juniper Networks Enterprise-Specific MIBs*

IPv6 MIB

- [Interpreting the Enterprise-Specific IPv6 MIB on page 507](#)
- [Standard IPv4/IPv6 MIBs on page 514](#)

Interpreting the Enterprise-Specific IPv6 MIB

- [IPv6 MIB on page 507](#)
- [jnxIpv6GlobalStats on page 507](#)
- [jnxIcmpv6GlobalStats on page 510](#)
- [jnxIpv6IfStatsTable on page 514](#)

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/junos13.2/topics/reference/mibs/mib-jnx-ipv6.txt.

For more information about the enterprise-specific IPv6 MIB objects, see the following topics:

- [jnxIpv6GlobalStats on page 507](#)
- [jnxIcmpv6GlobalStats on page 510](#)

jnxIpv6GlobalStats

The **jnxIpv6GlobalStats**, whose object ID is **{jnxIpv6Stats 1}**, represents the **jnxIpv6GlobalStats** branch of IPv6 stats. [Table 162 on page 507](#) identifies objects in the **jnxIpv6GlobalStats** branch.

Table 162: jnxIpv6GlobalStats

Object	Object ID	Description
jnxIpv6StatsReceives	jnxIpv6GlobalStats 1	The total number of IPv6 packets received.

Table 162: 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.

Table 162: 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 cannot 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 163 on page 509](#) identifies the count of all the other IPv6 next header types received.

Table 163: 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.
jnxIpv6StatsInIpls	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.

Table 163: 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.
jnxIpv6StatsInIv6s	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.
jnxIpv6StatsInFrgs	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.
jnxIpv6StatsInIsolps	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.

jnxIcmpv6GlobalStats

The **jnxIcmpv6GlobalStats**, whose object ID is **{jnxIpv6Stats 2}**, represents the jnxIcmpv6GlobalStats branch of IPv6 stats. [Table 164 on page 510](#) identifies objects in the jnxIcmpv6GlobalStats branch.

Table 164: jnxIcmpv6GlobalStats

Object	Object ID	Description
jnxIcmpv6StatsErrors	jnxIcmpv6GlobalStats 1	The total number of calls to icmp_error .

Table 164: jnxIcmpv6GlobalStats (*continued*)

Object	Object ID	Description
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.
jnxIcmpv6StatsOthers	jnxIcmpv6GlobalStats 20	The total number of generated message types other than the ones previously listed.

Table 164: jnxIcmpv6GlobalStats (*continued*)

Object	Object ID	Description
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 165 on page 512 identifies the count of all the other ICMPv6 message types received and transmitted.

Table 165: 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.
jnxIcmpv6StatsInRtrRenumbers	jnxIcmpv6GlobalStats 37	The total number of "router renumber" messages received.

Table 165: Count of All Other ICMPv6 Message Types Received and Transmitted (*continued*)

Object	Object ID	Description
jnxlcmpv6StatsInNIReqs	jnxlcmpv6GlobalStats 38	The total number of “node information request” messages received.
jnxlcmpv6StatsInNIReplies	jnxlcmpv6GlobalStats 39	The total number of “node information report” messages received.
jnxlcmpv6StatsOutUnreachables	jnxlcmpv6GlobalStats 40	The total number of “unreachable” messages transmitted.
jnxlcmpv6StatsOutPktTooBigs	jnxlcmpv6GlobalStats 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). jnxlcmpv6StatsTooBigs represents the number of 'too big packet' messages that actually goes out.
jnxlcmpv6StatsOutTimeExceeds	jnxlcmpv6GlobalStats 42	The total number of “time exceeded” messages transmitted.
jnxlcmpv6StatsOutParamProbs	jnxlcmpv6GlobalStats 43	The total number of “parameter problem” messages transmitted.
jnxlcmpv6StatsOutEchoReqs	jnxlcmpv6GlobalStats 44	The total number of “echo request” messages transmitted.
jnxlcmpv6StatsOutEchoReplies	jnxlcmpv6GlobalStats 45	The total number of “echo reply” messages transmitted.
jnxlcmpv6StatsOutMLQueries	jnxlcmpv6GlobalStats 46	The total number of “multicast listener query” messages transmitted.
jnxlcmpv6StatsOutMLReports	jnxlcmpv6GlobalStats 47	The total number of “multicast listener report” messages transmitted.
jnxlcmpv6StatsOutMLDones	jnxlcmpv6GlobalStats 48	The total number of “multicast listener done” messages transmitted.
jnxlcmpv6StatsOutRtrSolicits	jnxlcmpv6GlobalStats 49	The total number of “router solicitation” messages transmitted.
jnxlcmpv6StatsOutRtrAdvs	jnxlcmpv6GlobalStats 50	The total number of “router advertisement” messages transmitted.
jnxlcmpv6StatsOutNbrSolicits	jnxlcmpv6GlobalStats 51	The total number of “neighbor solicitation” messages transmitted.
jnxlcmpv6StatsOutNbrAdvs	jnxlcmpv6GlobalStats 52	The total number of “neighbor advertisement” messages transmitted.
jnxlcmpv6StatsOutRedirects	jnxlcmpv6GlobalStats 53	The total number of “redirect” messages transmitted.
jnxlcmpv6StatsOutRtrRenumbers	jnxlcmpv6GlobalStats 54	The total number of “router renumber” messages transmitted.

Table 165: Count of All Other ICMPv6 Message Types Received and Transmitted (*continued*)

Object	Object ID	Description
jnxIcmpv6StatsOutNIReqs	jnxIcmpv6GlobalStats 55	The total number of “node information request” messages transmitted.
jnxIcmpv6StatsOutNIReplys	jnxIcmpv6GlobalStats 56	The total number of “node information report” messages transmitted.

jnxIcmpv6IfStatsTable

The **jnxIcmpv6IfStatsTable**, whose object identifier is **{jnxIcmpv6IfStats 1}** contains the **jnxIcmpv6IfStatsEntry**. Each **jnxIcmpv6IfStatsEntry**, whose object identifier is **{jnxIcmpv6IfStatsTable 1}**, contains the objects listed in [Table 166 on page 514](#).

Table 166: jnxIcmpv6IfStatsEntry

Object	Object ID	Description
jnxIcmpv6IfInOctets	{jnxIcmpv6IfStatsEntry 1}	Indicates the total number of octets received on the interface.
jnxIcmpv6IfOutOctets	{jnxIcmpv6IfStatsEntry 2}	Indicates the total number of octets transmitted out of the interface.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Standard IPv4/IPv6 MIBs

The MIBs listed in this section are required to bring the system-wide statistics support in RFC 4293 for both IPv4 and IPv6 versions in Juniper Networks routers. With this support, the Internet service providers (ISPs) can manage IPv4 or IPv6 system-wide traffic statistics in a consistent manner. This feature keeps account of IPv6-specific packets and bytes traversing through the router. The feature can be used by ISPs to track the amount of IPv6 traffic going through their network compared to IPv4 traffic.

See [Table 158 on page 487](#) for a list of standard MIB groups implemented so that the ISPs can manage IPv4 or IPv6 system-wide traffic statistics in a consistent manner. See [Table 159 on page 502](#) for list of MIB objects supported by MX Series and EX Series devices. See [Table 160 on page 504](#) for list of supported Internet Control Message Protocol version 4 (ICMPv4) types and [Table 161 on page 505](#) for a list of supported ICMPv6 types by MX Series and EX Series devices.

For RFC information, go to: <http://www.ietf.org/rfc/rfc4293.txt> .

Table 167: Standard MIB Groups for IPv6

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipSystemStatsGroup	IP system wide statistics.	ipSystemStatsInReceives	The total number of input IP datagrams received, including those received in error. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInOctets	The total number of octets received in input IP datagrams, including those received in error. Octets from datagrams counted in ipSystemStatsInReceives <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime	{ipMIBGroups 8}
		ipSystemStatsInHdrErrors	The number of input IP datagrams discarded due to errors in their IP headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IP options, and so on. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInNoRoutes	The number of input IP datagrams discarded because no route could be found to transmit them to their destination. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInAddrErrors		{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The number of input IP datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, ::0). For entities that are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsInUnknownProtos	The number of locally addressed IP datagrams received successfully but discarded because of an unknown or unsupported protocol. When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInTruncatedPkts	The number of input IP datagrams discarded because the datagram frame did not carry enough data. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInFowDatagrams		{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful. When tracking interface statistics, the counter of the incoming interface is incremented for each datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsReasmReqds	The number of IP fragments received that needed to be reassembled at this interface. When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsReasmOKs	The number of IP datagrams successfully reassembled. When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsReasmFails		{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, and so on).</p> <p>NOTE: This is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received. When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	
		<code>ipSystemStatsInDiscards</code>	<p>The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but were discarded (for example, for lack of buffer space).</p> <p>NOTE: This counter does not include any datagrams discarded while awaiting re-assembly.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>
		<code>ipSystemStatsInDelivers</code>		<code>{ipMIBGroups 8}</code>

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of datagrams successfully delivered to IP user protocols (including ICMP). When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsOutRequests	<p>The total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission.</p> <p>NOTE: This counter does not include any datagrams counted in ipSystemStatsOutForwDatagrams.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.</p>	{ipMIBGroups 8}
		ipSystemStatsOutNoRoutes	<p>The number of locally generated IP datagrams discarded because no route could be found to transmit them to their destination.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.</p>	{ipMIBGroups 8}
		ipSystemStatsOutForwDatagrams		{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully forwarded datagram.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	
		<code>ipSystemStatsOutDiscards</code>	<p>The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but were discarded (for example, for lack of buffer space).</p> <p>NOTE: This counter would include datagrams counted in <code>ipSystemStatsOutForwDatagrams</code> if any such datagrams met this (discretionary) discard criterion.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>
		<code>ipSystemStatsOutFragReqds</code>	<p>The number of IP datagrams that would require fragmentation in order to be transmitted. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.</p> <p>Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ipSystemStatsDiscontinuityTime</code>.</p>	<code>{ipMIBGroups 8}</code>

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
		ipSystemStatsOutFragOKs	The number of IP datagrams that have been successfully fragmented. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutFragFails	The number of IP datagrams that have been discarded because they needed to be fragmented but could not be. This includes IPv4 packets that have the DF bit set and IPv6 packets that are being forwarded and exceed the outgoing link MTU. When tracking interface statistics, the counter of the outgoing interface is incremented for an unsuccessfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutFragCreates	The number of output datagram fragments that have been generated as a result of IP fragmentation. When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutTransmits		{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of IP datagrams that this entity supplied to the lower layers for transmission. This includes datagrams generated locally and those forwarded by this entity. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	
		ipSystemStatsOutOctets	The total number of octets in IP datagrams delivered to the lower layers for transmission. Octets from datagrams counted in ipSystemStatsOutTransmits <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInMcastPkts	The number of IP multicast datagrams received. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsInMcastOctets	The total number of octets received in IP multicast datagrams. Octets from datagrams counted in ipSystemStatsInMcastPkts <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutMcastPkts	The number of IP multicast datagrams transmitted. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsOutMcastOctets		

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The total number of octets transmitted in IP multicast datagrams. Octets from datagrams counted in ipSystemStatsOutMcastPkts <i>must</i> be counted here. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime.	{ipMIBGroups 8}
		ipSystemStatsDiscontinuityTime	The value of sysUpTime on the most recent occasion at which any one or more of this entry's counters suffered a discontinuity. If no such discontinuities have occurred since the last re- initialization of the local management subsystem, then this object contains a zero value.	{ipMIBGroups 8}
		ipSystemStatsRefreshRate	The minimum reasonable polling interval for this entry. This object provides an indication of the minimum amount of time required to update the counters in this entry.	{ipMIBGroups 8}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipAddressGroup	The group of objects for providing information about the addresses relevant to this entity's interfaces.	ipAddressSpinLock	<p>An advisory lock used to allow cooperating SNMP managers to coordinate their use of the set operation in creating or modifying rows within this table. In order to use this lock to coordinate the use of set operations, managers should first retrieve <code>ipAddressTableSpinLock</code>. They should then determine the appropriate row to create or modify. Finally, they should issue the appropriate set command, including the retrieved value of <code>ipAddressSpinLock</code>. If another manager has altered the table in the meantime, then the value of <code>ipAddressSpinLock</code> will have changed, and the creation will fail as it will be specifying an incorrect value for <code>ipAddressSpinLock</code>. It is suggested, but not required, that the <code>ipAddressSpinLock</code> be the first var bind for each set of objects representing a 'row' in a PDU.</p> <p>An agent is not required to provide write access to this object. However, if an agent provides write access to any of the other objects in the <code>ipAddressGroup</code>, it <i>should</i> provide write access to this object as well.</p>	{ipMIBGroups 19}
		ipAddressIfIndex	<p>The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's <code>ifIndex</code>.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressType	<p>The type of address.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressStatus		{ipMIBGroups 19}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The status of the address, describing if the address can be used for communication. In the absence of other information, an IPv4 address is always preferred.</p> <p>An agent is not required to provide write or create access to this object.</p>	
		ipAddressRowStatus	<p>The status of this conceptual row. The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified. A conceptual row cannot be made active until the ipAddressIfIndex has been set to a valid index.</p> <p>An agent is not required to provide write or create access to this object.</p>	{ipMIBGroups 19}
		ipAddressStorageType	<p>The storage type for this conceptual row. If this object has a value of 'permanent', then no other objects are required to be able to be modified.</p> <p>An agent is not required to provide write or create access to this object. If an agent allows this object to be written or created, it is not required to allow this object to be set to readOnly, permanent, or nonVolatile.</p>	{ipMIBGroups 19}
		ipAddressPrefix	<p>A pointer to the row in the prefix table to which this address belongs. Can be { 0 0 } if there is no such row.</p>	{ipMIBGroups 19}
		ipAddressOrigin	<p>The origin of the address.</p>	{ipMIBGroups 19}
		ipAddressCreated	<p>The value of sysUpTime at the time this entry was created. If this entry was created prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.</p>	{ipMIBGroups 19}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
		ipAddressLastChanged	The value of sysUpTime at the time this entry was last updated. If this entry was updated prior to the last re- initialization of the local network management subsystem, then this object contains a zero value.	{ipMIBGroups 19}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
ipNetToPhysicalGroup	The group of objects for providing information about the mappings of the network address to the physical address known to this node.	ipNetToPhysicalPhysAddress	The media-dependent 'physical' address. As the entries in this table are typically not persistent when this object is written, the entity <i>should not</i> save the change to non volatile storage. An agent is not required to provide write or create access to this object.	{ipMIBGroups 20}
		ipNetToPhysicalLastUpdated	The value of sysUpTime at the time this entry was last updated. If this entry was updated prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.	{ipMIBGroups 20}
		ipNetToPhysicalType		{ipMIBGroups 20}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			<p>The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToPhysicalTable. That is, it effectively disassociates the interface identified with that entry from the mapping identified with that entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToPhysicalType object. The 'dynamic(3)' type indicates that the IP address to physical addresses mapping has been dynamically resolved using for example, IPv4 ARP or the IPv6 Neighbor Discovery protocol. The 'static(4)' type indicates that the mapping has been statically configured. Both of these types refer to entries that provide mappings for other entities' addresses. The 'local(5)' type indicates that the mapping is provided for an entity's own interface address. As the <i>entries</i> in this table are typically not persistent when this object is written, the entity <i>should not</i> save the change to nonvolatile storage.</p> <p>An agent is not required to provide write or create access to this object.</p>	
		ipNetToPhysicalState	The Neighbor Unreachability Detection state for the interface when the address mapping in this entry is used.	{ipMIBGroups 20}
		ipNetToPhysicalRowStatus		{ipMIBGroups 20}

Table 167: Standard MIB Groups for IPv6 (*continued*)

Group	Description/ Notes	Objects	Description/Notes	Object Identifier
			The status of this conceptual row. The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified. A conceptual row cannot be made active until the ipNetToPhysicalPhysAddress object has been set. Note that if the ipNetToPhysicalType is set to 'invalid', the managed node might delete the entry independent of the state of this object.	
icmpStatsGroup	The group of objects providing ICMP statistics.	icmpStatsInMsgs	The total number of ICMP messages that the entity received. Note that this counter includes all those counted by icmpStatsInErrors.	{ipMIBGroups 24}
		icmpStatsInErrors	The number of ICMP messages that the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, and so on).	{ipMIBGroups 24}
		icmpStatsOutMsgs	The total number of ICMP messages that the entity attempted to send. Note that this counter includes all those counted by icmpStatsOutErrors.	{ipMIBGroups 24}
		icmpStatsOutErrors	The number of ICMP messages that this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value should not include errors discovered outside the ICMP layer, such as the inability of IP to route the resultant datagram. In some implementations, there might be no types of error that contribute to this counter's value.	{ipMIBGroups 24}
		icmpMsgStatsInPkts	The number of input packets for this AF and type.	{ipMIBGroups 24}
		icmpMsgStatsOutPkts	The number of output packets for this AF and type.	{ipMIBGroups 24}

Table 168: MIB Objects Supported by MX Series and EX Series

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
ipSystemStatsInReceives	Yes	Yes
ipSystemStatsInOctets	Yes	No
ipSystemStatsInHdrErrors	Yes	No
ipSystemStatsInNoRoutes	Yes	Yes
ipSystemStatsInAddrErrors	Yes	No
ipSystemStatsInUnknownProtos	Yes	Yes
ipSystemStatsInTruncatedPkts	Yes	No
ipSystemStatsInForwDatagrams	Yes	Yes
ipSystemStatsReasmReqds	Yes	Yes
ipSystemStatsReasmOKs	Yes	Yes
ipSystemStatsReasmFails	Yes	Yes
ipSystemStatsInDiscards	Yes	No
ipSystemStatsInDelivers	Yes	Yes
ipSystemStatsOutRequests	Yes	Yes
ipSystemStatsOutNoRoutes	Yes	Yes
ipSystemStatsOutForwDatagrams	Yes	Yes
ipSystemStatsOutDiscards	Yes	No
ipSystemStatsOutFragReqds	Yes	No
ipSystemStatsOutFragOKs	Yes	Yes
ipSystemStatsOutFragFails	Yes	Yes
ipSystemStatsOutFragCreates	Yes	Yes
ipSystemStatsOutTransmits	Yes	Yes
ipSystemStatsOutOctets	Yes	No
ipSystemStatsInMcastPkts	Yes	No

Table 168: MIB Objects Supported by MX Series and EX Series (*continued*)

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
ipSystemStatsInMcastOctets	Yes	No
ipSystemStatsOutMcastPkts	Yes	No
ipSystemStatsOutMcastOctets	Yes	No
ipSystemStatsDiscontinuityTime	Yes	No
ipSystemStatsRefreshRate	Yes	No
ipAddressSpinLock	Yes	No
ipAddressIfIndex	Yes	No
ipAddressType	Yes	No
ipAddressOrigin	Yes	No
ipAddressStatus	Yes	No
ipAddressCreated	Yes	Yes
ipAddressLastChanged	Yes	Yes
ipAddressRowStatus	Yes	Yes
ipAddressStorageType	Yes	No
ipNetToPhysicalPhysAddress	Yes	No
ipNetToPhysicalLastUpdated	Yes	Yes
ipNetToPhysicalType	Yes	No
ipNetToPhysicalState	Yes	Yes
ipNetToPhysicalRowStatus	Yes	No
icmpStatsInMsgs	Yes	Yes
icmpStatsInErrors	Yes	Yes
icmpStatsOutMsgs	Yes	Yes
icmpStatsOutErrors	Yes	Yes
icmpMsgStatsInPkts	Yes	Yes

Table 168: MIB Objects Supported by MX Series and EX Series (*continued*)

MIB Objects	Supported by MX Series? Yes/No	Supported by EX Series? Yes/No
icmpMsgStatsOutPkts	Yes	Yes

Table 169: Supported ICMPv4 Counters

Counter Type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
0	Echo Reply	Yes	Yes
3	Destination Unreachable	Yes	Yes
4	Source Quench (Deprecated)	Yes	No
5	Redirect	Yes	Yes
6	Alternate Host Address	Yes	No
8	Echo	Yes	Yes
9	Router Advertisement	Yes	No
10	Router Selection	Yes	No
11	Time Exceeded	Yes	Yes
12	Parameter Problem	Yes	Yes
13	Timestamp	Yes	No
14	Timestamp Reply	Yes	Yes
15	Information Request	Yes	No
16	Information Reply	Yes	No
17	Address Mask Request	Yes	No
18	Address Mask Reply	Yes	No
30	Traceroute	Yes	No
31	Datagram Conversion Error	Yes	No
32	Mobile Host Redirect	Yes	—
33	IPv6 Where-Are-You	Yes	—
34	IPv6 I-Am-Here	Yes	—

Table 169: Supported ICMPv4 Counters (*continued*)

Counter Type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
35	Mobile Registration Request	Yes	–
36	Mobile Registration Reply	Yes	–
39	SKIP	Yes	–
40	Photuris	Yes	–

Table 170: Supported ICMPv6 Counters

Counter type	Name	icmpMsgStatsInPkts	icmpMsgStatsOutPkts
1	Destination Unreachable	Yes	Yes
2	Packet Too Big	Yes	No
3	Time Exceeded	Yes	No
4	Parameter Problem	Yes	Yes
128	Echo Request	Yes	Yes
129	Echo Reply	Yes	Yes
130	Multicast Listener Query	Yes	Yes
131	Multicast Listener Report	Yes	Yes
132	Multicast Listener Done	Yes	Yes
133	Router Solicitation	Yes	–
134	Router Advertisement	Yes	Yes
135	Neighbor Solicitation	Yes	Yes
136	Neighbor Advertisement	Yes	Yes
137	Redirect Message	Yes	No
138	Router Renumbering	Yes	–
139	ICMP Node Information Query	Yes	–
140	ICMP Node Information Response	Yes	–
143	V2 Multicast Listener Report	Yes	–



NOTE: The conventions (Yes, No, –) indicate the following information:

- Yes: Supported (Junos OS supports statistics for the packet type)
- No: Not supported (Junos OS does not support statistics for the packet type)
- – : Not applicable (Junos OS does not support the packet type)

**Related
Documentation**

- *Juniper Networks Enterprise-Specific MIBs*

CHAPTER 40

L2ALD MIB

- [Interpreting the Enterprise-Specific L2ALD MIB on page 537](#)

Interpreting the Enterprise-Specific L2ALD MIB

- [L2ALD MIB on page 537](#)
- [jnxl2aldInterfaceTable on page 537](#)
- [MAC Address Limit Traps on page 538](#)

L2ALD MIB

The Juniper Networks enterprise-specific Layer 2 Address Learning Daemon (L2ALD) MIB, whose object identifier is **{jnxl2aldMibRoot 1}**, contains information about Layer 2 addresses and defines L2ALD traps.

The L2ALD MIB has the following two branches:

- **jnxl2aldNotification**, whose object identifier is **{jnxl2aldMib 0}**
- **jnxl2aldObjects**, whose object identifier is **{jnxl2aldMib 1}**

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2ald.txt.

For information about the enterprise-specific L2ALD MIB objects, see the following topics:

- [jnxl2aldInterfaceTable on page 537](#)
- [MAC Address Limit Traps on page 538](#)

jnxl2aldInterfaceTable

The **jnxl2aldInterfaceTable**, whose object identifier is **{jnxl2aldObjects 1}**, contains objects that control the MAC address limit on each interface. Each **jnxl2aldEntry** (object identifier: **{jnxl2aldInterfaceTable 1}**) in the **jnxl2aldInterfaceTable** contains the objects listed in [Table 171 on page 538](#).

Table 171: jnxl2aldInterfaceTable

Object	Object ID	Description
jnxl2aldIntfLogicalRouter	jnxl2aldEntry 1	Identifies the logical system with which the jnxl2aldEntry is associated.
jnxl2aldIntfRoutingInst	jnxl2aldEntry 2	Identifies the routing instance with which the jnxl2aldEntry is associated.
jnxl2aldIntfBridgeDomain	jnxl2aldEntry 3	Identifies the bridge domain with which the jnxl2aldEntry is associated.
jnxl2aldIntfMacLimit	jnxl2aldEntry 4	Defines the MAC address limit for the physical interface associated with the jnxl2aldEntry .
Scalar Objects for Notifications		
jnxl2aldRoutingInst	jnxl2aldObjects 2	Specifies the routing instance for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldBridgeDomain	jnxl2aldObjects 3	Specifies the bridge domain for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldLogicalRouter	jnxl2aldObjects 4	Specifies the logical system for the jnxl2aldRoutingInstMacLimit trap.
jnxl2aldMacLimit	jnxl2aldObjects 5	Specifies the maximum number of MAC addresses that can be learned by the routing instance.
jnxl2aldGbMacLimit	jnxl2aldObjects 6	Specifies the maximum number of MAC addresses that can be learned by the router.

MAC Address Limit Traps

The enterprise-specific L2ALD MIB defines the following traps:

- **jnxl2aldRoutingInstMacLimit**, whose object identifier is **{jnxl2aldNotification 1}**, is generated when the number of MAC addresses for the given routing instance, **jnxl2aldRoutingInst**, exceeds the set limit. This trap contains the following objects: **jnxl2aldLogicalRouter**, **jnxl2aldRoutingInst**, **jnxl2aldBridgeDomain**, and **jnxl2aldMacLimit**.
- **jnxl2aldInterfaceMacLimit**, whose object identifier is **{jnxl2aldNotification 2}**, is generated when the number of MAC addresses for the given physical interface exceeds the set limit. This trap contains the following objects: **jnxl2aldIntfLogicalRouter**, **jnxl2aldIntfRoutingInst**, **jnxl2aldIntfBridgeDomain**, **ifDescr**, and **jnxl2aldIntfMacLimit**.
- **jnxl2aldGlobalMacLimit**, whose object identifier is **{jnxl2aldNotification 3}**, is generated when the MAC limit for the entire system exceeds the set limit.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)

- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

L2CP MIB

- Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB on page 541

Interpreting the Enterprise-Specific Layer 2 Control Protocol (L2CP) MIB

- L2CP MIB on page 541
- L2CP MIB Objects Supported by the Junos OS on page 541

L2CP MIB

The Juniper Networks enterprise-specific Layer 2 Control Protocol (L2CP) MIB, whose object identifier is {**jnxMibs 53**}, provides information about LC2P-based features on MX Series 3D Universal Edge Routers . Currently, the 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/junos13.2/topics/reference/mibs/mib-jnx-l2cp-features.txt .

For more information about the enterprise-specific L2CP MIB objects supported by the Junos OS, see the following topic:

- L2CP MIB Objects Supported by the Junos OS on page 541

L2CP MIB Objects Supported by the Junos OS

Table 172 on page 541 lists the L2CP MIB objects supported by the Junos OS.

Table 172: L2CP MIB Objects Supported by the Junos OS

Object	Object ID	Description
jnxDot1dStpPortRootProtectEnabled	jnxDot1dStpPortProtectEntry 1	Indicates whether the root protect functionality is enabled on the port. If set to true , the port cannot be selected as the root port even if it has the best spanning-tree priority value. By default this object is set to false .

Table 172: L2CP MIB Objects Supported by the Junos OS (*continued*)

Object	Object ID	Description
<code>jnxDot1dStpPortRootProtectState</code>	<code>jnxDot1dStpPortProtectEntry</code> 2	<p>Returns one of the following integer values to indicate whether the port was ever prevented from being the root port or not:</p> <ul style="list-style-type: none"> • 0 no-error—Indicates that the port was not prevented from being a root port. • 1 root-prevented—Indicates that the port was prevented from being a root port. <p>This object always indicates a 0 no-error state if the <code>jnxDot1dStpPortRootProtectEnabled</code> is set to false.</p>
<code>jnxPortRootProtectStateChangeTrap</code>	<code>jnxL2cpProtectTraps</code> 1	Generated when there is a change in the <code>jnxDot1dStpPortRootProtectState</code> for a port.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 42

L2TP MIB

- [Interpreting the Enterprise-Specific L2TP MIB on page 543](#)

Interpreting the Enterprise-Specific L2TP MIB

- [L2TP MIB on page 543](#)
- [L2TP Scalar Status and Statistics Group on page 543](#)
- [jnxL2tpTunnelGroupStatsTable on page 544](#)
- [jnxL2tpTunnelStatsTable on page 545](#)
- [jnxL2tpSessionStatsTable on page 547](#)
- [jnxL2tpMlpppBundleStatsTable on page 554](#)

L2TP MIB

The Juniper Networks enterprise-specific Layer 2 Tunneling Protocol (L2TP) MIB enables you to monitor L2TP tunnels and sessions using SNMP on both M Series and MX Series routers. The MX Series router uses the Common Edge L2TP process, jlt2pd. Some L2TP MIB objects are not supported by jlt2pd. If an object is not supported, the object returns either zero or the default value. The L2TP MIB, whose object identifier is **{jnxMibs 49}**, provides information related to L2TP tunnels and sessions.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-l2tp.txt.

For information about the enterprise-specific L2TP MIB objects, see the following topics:

- [L2TP Scalar Status and Statistics Group on page 543](#)
- [jnxL2tpTunnelGroupStatsTable on page 544](#)
- [jnxL2tpTunnelStatsTable on page 545](#)
- [jnxL2tpSessionStatsTable on page 547](#)
- [jnxL2tpMlpppBundleStatsTable on page 554](#)

L2TP Scalar Status and Statistics Group

[Table 173 on page 544](#) lists the objects in the L2TP scalar status and statistics group.

Table 173: L2TP Scalar Status and Statistics Group

Object	Object Identifier	Description
<code>jnxL2tpStatsTotalTunnels</code>	<code>jnxL2tpStats 1</code>	Returns the total number of tunnels that are in established state at the time of querying.
<code>jnxL2tpStatsTotalSessions</code>	<code>jnxL2tpStats 2</code>	Returns the total number of sessions that are in established state at the time of querying.
<code>jnxL2tpStatsControlRxOctets</code>	<code>jnxL2tpStats 3</code>	Returns the number of control channel octets received by the tunnels in established state at the time of querying.
<code>jnxL2tpStatsControlRxPkts</code>	<code>jnxL2tpStats 4</code>	Returns the number of control packets received by the tunnels in established state at the time of querying.
<code>jnxL2tpStatsControlTxOctets</code>	<code>jnxL2tpStats 5</code>	Returns the number of control channel octets that were transmitted to tunnel endpoints that are in established state at the time of querying.
<code>jnxL2tpStatsControlTxPkts</code>	<code>jnxL2tpStats 6</code>	Returns the number of control packets that were transmitted to the tunnel endpoints that are in established state at the time of querying.
<code>jnxL2tpStatsPayloadRxPkts</code>	<code>jnxL2tpStats 8</code>	Returns the number of payload packets that were received on the tunnels that are in established state at the time of querying.
<code>jnxL2tpStatsPayloadTxOctets</code>	<code>jnxL2tpStats 9</code>	Returns the number of payload channel octets that were transmitted to the tunnel peers that are in established state at the time of querying.
<code>jnxL2tpStatsPayloadTxPkts</code>	<code>jnxL2tpStats 10</code>	Returns the number of payload packets that were transmitted to existing tunnel peers that are in established state at the time of querying.
<code>jnxL2tpStatsErrorTxPkts</code>	<code>jnxL2tpStats 11</code>	Returns the number of packet transmission attempts with errors to the tunnel peers that are in established state at the time of querying.
<code>jnxL2tpStatsErrorRxPkts</code>	<code>jnxL2tpStats 12</code>	Returns the number of packets with errors that were received from the existing tunnel peers that are in established state at the time of querying.
<code>jnxL2tpStatsPayloadRxOctets64</code>	<code>jnxL2tpStats 13</code>	Returns the number of payload channel octets that were received on the tunnels that are in established state at the time of querying.

`jnxL2tpTunnelGroupStatsTable`

The `jnxL2tpTunnelGroupStatsTable`, whose object ID is `{jnxL2tpObjects 2}`, contains objects that describe the current status and statistics of an L2TP tunnel group.

The `jnxL2tpTunnelGroupStatsEntry` objects are listed in [Table 174 on page 545](#).

Table 174: jnxL2tpTunnelGroupStatsTable

Object	Object Identifier	Description
jnxL2tpTunnelGroupStatsTnlGrpName	jnxL2tpTunnelGroupStatsEntry 1	The name of the particular tunnel group.
jnxL2tpTunnelGroupStatsGatewayAddrType	jnxL2tpTunnelGroupStatsEntry 2	The type of local IP address for L2TP tunnels that are part of the group.
jnxL2tpTunnelGroupStatsGatewayAddr	jnxL2tpTunnelGroupStatsEntry 3	The local IP address for L2TP tunnels that are part of the group.
jnxL2tpTunnelGroupStatsSvcIntfName	jnxL2tpTunnelGroupStatsEntry 4	The name of the service interface that is hosting the tunnel group.
jnxL2tpTunnelGroupStatsTotalTunnels	jnxL2tpTunnelGroupStatsEntry 5	The total number of tunnels that are in the established state at the time of querying.
jnxL2tpTunnelGroupStatsTotalSessions	jnxL2tpTunnelGroupStatsEntry 6	The total number of established sessions in the tunnel group at the time of querying.

jnxL2tpTunnelStatsTable

jnxL2tpTunnelStatsTable, whose object ID is **{jnxL2tpObjects 3}**, contains objects that describe the current status and statistics of an L2TP tunnel.

A **jnxL2tpTunnelStatsEntry** represents an L2TP tunnel interface statistics entry and has objects that are listed in [Table 175 on page 545](#).

Table 175: jnxL2tpTunnelStatsTable

Object	Object Identifier	Description
jnxL2tpTunnelStatsLocalTID	jnxL2tpTunnelStatsEntry 1	The local tunnel Identifier.
jnxL2tpTunnelStatsServiceInterface	jnxL2tpTunnelStatsEntry 2	The name of the service interface on which the tunnel is being hosted. NOTE: This attribute is not supported on jl2tpd LAC.
jnxL2tpTunnelStatsTunnelGroup	jnxL2tpTunnelStatsEntry 3	The name of the tunnel group to which the tunnel belongs. NOTE: This attribute is not supported on jl2tpd LAC.
jnxL2tpTunnelStatsRemoteTID	jnxL2tpTunnelStatsEntry 4	The remote tunnel identifier. See RFC 2661, Section 3.1.
jnxL2tpTunnelStatsRemoteIpAddrType	jnxL2tpTunnelStatsEntry 5	The type of the remote-end address of the tunnel.
jnxL2tpTunnelStatsRemoteIpAddress	jnxL2tpTunnelStatsEntry 6	The remote-end address of the tunnel.
jnxL2tpTunnelStatsRemoteUdpPort	jnxL2tpTunnelStatsEntry 7	The remote-end UDP port of the tunnel.

Table 175: jnxL2tpTunnelStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpTunnelStatsActiveSessions	jnxL2tpTunnelStatsEntry 8	The total number of sessions that are in established state for the tunnel.
jnxL2tpTunnelStatsStat	jnxL2tpTunnelStatsEntry 9	<p>One of the following states for the control tunnel:</p> <ul style="list-style-type: none"> cc_responder_accept_new—Shows that the tunnel has received and accepted the start control connection request (SCCRQ). cc_responder_reject_new—Shows that the tunnel has received and rejected the SCCRQ. cc_responder_idle—Shows that the tunnel has just been created. cc_responder_wait_ctl_conn—Shows that the tunnel has sent the start control connection response (SCCRP) and is waiting for the start control connection connected (SCCCN) message. Cleanup—Shows that the tunnel is being cleaned up. Closed—Shows that the tunnel is being closed. Destroyed—Shows that the tunnel is being destroyed. Established—Shows that the tunnel is operational. Terminate—Shows that the tunnel is being terminated. Unknown—Shows that the tunnel is not connected to the router.
jnxL2tpTunnelStatsLocalIpAddrType	jnxL2tpTunnelStatsEntry 10	The type of local-end address of the tunnel.
jnxL2tpTunnelStatsLocalIpAddress	jnxL2tpTunnelStatsEntry 11	The local-end address of the tunnel.
jnxL2tpTunnelStatsLocalUdpPort	jnxL2tpTunnelStatsEntry 12	The local-end UDP port number of the tunnel.
jnxL2tpTunnelStatsLocalHostName	jnxL2tpTunnelStatsEntry 13	The local hostname of the tunnel.
jnxL2tpTunnelStatsRemoteHostName	jnxL2tpTunnelStatsEntry 14	The hostname of the L2TP peer, as discovered during the tunnel establishment phase (using the Host Name AVP). If the tunnel is idle, this object shows the value based on the data from the last time it was connected.
jnxL2tpTunnelMaxSessions	jnxL2tpTunnelStatsEntry 15	The maximum number of sessions configured on the tunnel. Value could be a positive number or zero (unlimited).
jnxL2tpTunnelStatsWindowSize	jnxL2tpTunnelStatsEntry 16	The send window size for the tunnel.
jnxL2tpTunnelStatsHelloInterval	jnxL2tpTunnelStatsEntry 17	The hello interval for the tunnel.
jnxL2tpTunnelStatsCreationTime	jnxL2tpTunnelStatsEntry 18	The time when the tunnel was created.
jnxL2tpTunnelStatsUpTime	jnxL2tpTunnelStatsEntry 19	The time elapsed since the tunnel was established.

Table 175: jnxL2tpTunnelStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpTunnelStatsIdleTime	jnxL2tpTunnelStatsEntry 20	The time elapsed since the last data activity, transmission or reception, on the tunnel.
jnxL2tpTunnelStatsCollectionStart	jnxL2tpTunnelStatsEntry 21	The time at which the statistics gathering started for the tunnel.
jnxL2tpTunnelStatsControlTxPkts	jnxL2tpTunnelStatsEntry 22	The number of control packets transmitted to the tunnel peer.
jnxL2tpTunnelStatsControlRxPkts	jnxL2tpTunnelStatsEntry 24	The number of control packets received on the tunnel.
jnxL2tpTunnelStatsDataTxBytes	jnxL2tpTunnelStatsEntry 27	The number of data bytes transmitted to the tunnel peer.
jnxL2tpTunnelStatsDataRxBytes	jnxL2tpTunnelStatsEntry 29	The number of data bytes received from the tunnel peer.
jnxL2tpTunnelStatsErrorTxPkts	jnxL2tpTunnelStatsEntry 30	The number of error transmit packets on the tunnel. <i>NOTE:</i> This attribute is not available on Trio MPC/MICs.
jnxL2tpTunnelStatsErrorRxPkts	jnxL2tpTunnelStatsEntry 31	The number of error receive packets on the tunnel. <i>NOTE:</i> This attribute is not available on Trio MPC/MICs.
jnxL2tpTunnelStatsControlTxBytes32	jnxL2tpTunnelStatsEntry 32	The number of control bytes transmitted to the tunnel peer.
jnxL2tpTunnelStatsControlRxBytes32	jnxL2tpTunnelStatsEntry 33	The number of control bytes received from the tunnel peer.
jnxL2tpTunnelStatsDataTxPkts64	jnxL2tpTunnelStatsEntry 34	The number of data packets transmitted to the tunnel.
jnxL2tpTunnelStatsDataRxPkts64	jnxL2tpTunnelStatsEntry 35	The number of data packets received from the tunnel.

jnxL2tpSessionStatsTable

jnxL2tpSessionStatsTable, whose object ID is **{jnxL2tpObjects 4}**, contains the objects that describe the current status and statistics of a single L2TP tunneled session.

A **jnxL2tpSessionStatsEntry** represents an L2TP session interface status and has the objects that are listed in [Table 176 on page 547](#).

Table 176: jnxL2tpSessionStatsTable

Object	Object Identifier	Description
jnxL2tpSessionStatsLocalTID	jnxL2tpSessionStatsEntry 1	The local tunnel Identifier.

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsLocalSID	jnxL2tpSessionStatsEntry 2	The local session Identifier.
jnxL2tpSessionStatsServiceInterface	jnxL2tpSessionStatsEntry 3	<p>The name of the service interface on which this session is being hosted.</p> <p>NOTE: This attribute is not supported on jl2tpd LAC.</p>
jnxL2tpSessionStatsTunnelGroup	jnxL2tpSessionStatsEntry 4	<p>The name of the tunnel group to which this session belongs.</p> <p>NOTE: This attribute is not supported on jl2tpd LAC.</p>
jnxL2tpSessionStatsRemoteSID	jnxL2tpSessionStatsEntry 5	The remote-end assigned session identifier for this session. This value remains zero from the time of starting the session until the time the remote end point responds.
jnxL2tpSessionStatsInterfaceUnit	jnxL2tpSessionStatsEntry 6	The interface unit number that corresponds to the logical service interface on which the session is being hosted.
jnxL2tpSessionStatsEncapType	jnxL2tpSessionStatsEntry 7	The tunnel encapsulation type.
jnxL2tpSessionStatsBundleID	jnxL2tpSessionStatsEntry 8	The ID of the bundle to which the session is linked. This field is valid only for tunnel encapsulation type multilink-ppp .

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsStat	jnxL2tpSessionStatsEntry 9	<p>One of the following status messages to show the state of the session at the time of querying:</p> <ul style="list-style-type: none"> Established— The session is operational. Closed—The session has been closed. Destroyed—The session has been destroyed. Cleanup—The session has been cleaned up. Ins_ic_accept_new—The new session has been accepted. Ins_ic_idle—The session has been created but is in idle state. Ins_ic_reject_new—A new session has been rejected. Ins_ic_wait_connect—The session is waiting for the peer's incoming call connected (ICCN) message.
jnxL2tpSessionStatsUserName	jnxL2tpSessionStatsEntry 10	<p>The peer session name on the interface. This is typically the login name of the remote user. This object contains a null string when the user name is unknown to the local tunnel peer.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsMode	jnxL2tpSessionStatsEntry 11	The configured mode value for this session.
jnxL2tpSessionStatsLocalAddrType	jnxL2tpSessionStatsEntry 12	The type of the local-end address of the tunnel that hosts the session.
jnxL2tpSessionStatsLocalAddress	jnxL2tpSessionStatsEntry 13	The local end address of the tunnel that hosts the session.
jnxL2tpSessionStatsLocalUdpPort	jnxL2tpSessionStatsEntry 14	The UDP port of the local end of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteAddrType	jnxL2tpSessionStatsEntry 15	The type of the remote end address of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteAddress	jnxL2tpSessionStatsEntry 16	The remote end address of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteUdpPort	jnxL2tpSessionStatsEntry 17	The UDP port of the remote-end of the tunnel that hosts the session.

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsLocalHostName	jnxL2tpSessionStatsEntry 18	The local hostname of the tunnel that hosts the session.
jnxL2tpSessionStatsRemoteHostName	jnxL2tpSessionStatsEntry 19	The hostname as discovered during the tunnel establishment phase (using the Host Name AVP) of the L2TP peer.
jnxL2tpSessionAssignedIpAddrType	jnxL2tpSessionStatsEntry 20	<p>The type of IP address of PPP client being tunneled as obtained from IPCP configuration while establishing the session.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionAssignedIpAddress	jnxL2tpSessionStatsEntry 21	<p>The IP address of the PPP client being tunneled as obtained from IPCP configuration while establishing the session.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionLocalMRU	jnxL2tpSessionStatsEntry 22	<p>The MRU for the local PPP entity. This value is the MRU that the remote entity uses when sending packets to the session.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionRemoteMRU	jnxL2tpSessionStatsEntry 23	<p>The MRU for the remote PPP entity. This value is the MRU that the local entity uses when sending packets to the remote PPP client.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsTxSpeed	jnxL2tpSessionStatsEntry 24	The last known transmit baud rate for the session.
jnxL2tpSessionStatsRxSpeed	jnxL2tpSessionStatsEntry 25	The last known receive baud rate for the session.
jnxL2tpSessionStatsCallBearerType	jnxL2tpSessionStatsEntry 26	The bearer type of this session.
jnxL2tpSessionStatsFramingType	jnxL2tpSessionStatsEntry 27	The framing type of the session.
jnxL2tpSessionStatsLCPRenegotiation	jnxL2tpSessionStatsEntry 28	The ON/OFF state of the LCP renegotiation for the session.

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsAuthMethod	jnxL2tpSessionStatsEntry 29	<p>The proxy authentication method employed by the LAC for the session.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsNasIpAddrType	jnxL2tpSessionStatsEntry 30	<p>The type of IP address of the RADIUS network address server to which the accounting records for this session are being sent.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsNasIpAddress	jnxL2tpSessionStatsEntry 31	<p>The IP address of the RADIUS network address server to which the accounting records for the session are being sent.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsNasIpPort	jnxL2tpSessionStatsEntry 32	<p>The port on which RADIUS network address server accounting messages are sent.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsFramedProtocol	jnxL2tpSessionStatsEntry 33	<p>The frame protocol attribute obtained from the RADIUS server for the session.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsFramedIpAddrType	jnxL2tpSessionStatsEntry 34	<p>The IP address to be configured for the user, as provided by the RADIUS server in response to authentication request.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>
jnxL2tpSessionStatsFramedIpAddress	jnxL2tpSessionStatsEntry 35	<p>The IP address to be configured for the user, as provided by the RADIUS server in response to the authentication request.</p> <p>NOTE: This attribute is not supported on jl2tpd.</p>

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsCallingStationID	jnxL2tpSessionStatsEntry 36	The phone number that the call came from. The RADIUS NAS obtains the phone number that the call came from by using Automatic Number Identification (ANI) or similar technology. It is used only in access-request packets.
jnxL2tpSessionStatsCalledStationID	jnxL2tpSessionStatsEntry 37	The phone number that the user called. The RADIUS NAS obtains the phone number that the user called by using Dialed Number Identification Service (DNIS) or similar technology. It is used only in Access-Request packets.
jnxL2tpSessionStatsAcctDelayTime	jnxL2tpSessionStatsEntry 38	The duration (in seconds) for which the RADIUS accounting client has been trying to send a record for. This value can be subtracted from the time of arrival on the server to find the approximate time of the event generating this Accounting-Request. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctSessionID	jnxL2tpSessionStatsEntry 39	A unique Accounting ID to help match start and stop records in a log file. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctMethod	jnxL2tpSessionStatsEntry 40	The accounting method employed for this session. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctSessionTime	jnxL2tpSessionStatsEntry 41	The number of seconds for which the user has received service. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctNasPortType	jnxL2tpSessionStatsEntry 42	The type of the physical port of the NAS that performs accounting for the user. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctTnlClientEndPoint	jnxL2tpSessionStatsEntry 43	This object contains the remote tunnel identifier of the tunnel that hosts the session.

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsAcctTnlServerEndPoint	jnxL2tpSessionStatsEntry 44	The local tunnel identifier of the tunnel that hosts the session.
jnxL2tpSessionStatsAcctTnlClientAuthID	jnxL2tpSessionStatsEntry 45	The remote hostname of the tunnel that hosts the session as discovered during the tunnel establishment phase (using the Host Name AVP) of the L2TP peer. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsAcctTnlServerAuthID	jnxL2tpSessionStatsEntry 46	The local hostname of the tunnel that hosts the session. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsUserProfileName	jnxL2tpSessionStatsEntry 47	The configured access profile name that is being used for the session. NOTE: This attribute is not supported on jl2tpd.
jnxL2tpSessionStatsInterfaceID	jnxL2tpSessionStatsEntry 48	The interface identification (name) for the service interface that bears the session. NOTE: This attribute is not supported on jl2tpd LAC.
jnxL2tpSessionStatsCallSerialNumber	jnxL2tpSessionStatsEntry 49	The serial number assigned to the session.
jnxL2tpSessionStatsCreationTim	jnxL2tpSessionStatsEntry 50	Time when the session was created.
jnxL2tpSessionStatsUpTim	jnxL2tpSessionStatsEntry 51	The time elapsed since the session was established.
jnxL2tpSessionStatsIdleTime	jnxL2tpSessionStatsEntry 52	The time elapsed since the session had any data activity (transmission or reception).
jnxL2tpSessionStatsCollectionStart	jnxL2tpSessionStatsEntry 53	The time at which the statistics gathering started for the session.
jnxL2tpSessionStatsControlTxPkts	jnxL2tpSessionStatsEntry 54	The number of control packets transmitted to the session peer.
jnxL2tpSessionStatsControlRxPkts	jnxL2tpSessionStatsEntry 56	The number of control packets received on the session.

Table 176: jnxL2tpSessionStatsTable (*continued*)

Object	Object Identifier	Description
jnxL2tpSessionStatsDataTxBytes	jnxL2tpSessionStatsEntry 59	The number of data bytes transmitted to the session peer.
jnxL2tpSessionStatsDataRxBytes	jnxL2tpSessionStatsEntry 61	The number of data bytes received from the session peer.
jnxL2tpSessionStatsErrorTxPkt	jnxL2tpSessionStatsEntry 62	The number of error transmit packets on the session.
jnxL2tpSessionStatsErrorRxPkts	jnxL2tpSessionStatsEntry 63	The number of error receive packets on the session.
jnxL2tpSessionStatsControlTxBytes32	jnxL2tpSessionStatsEntry 64	The number of control bytes that were transmitted to the session peer.
jnxL2tpSessionStatsControlRxBytes32	jnxL2tpSessionStatsEntry 65	The number of control bytes received from the session peer.
jnxL2tpSessionStatsDataTxPkts64	jnxL2tpSessionStatsEntry 66	The number of data packets transmitted to the remote session peer.
jnxL2tpSessionStatsDataRxPkts64	jnxL2tpSessionStatsEntry 67	The number of data packets received on this session.

jnxL2tpMlpppBundleStatsTable

The **jnxL2tpMlpppBundleStatsTable**, whose object ID is **{jnxL2tpObjects 5}**, contains objects that describe the current status and statistics of a single L2TP tunneled multilink PPP bundle.

A **jnxL2tpMlpppBundleStatsEntry** represents the L2TP MLPPP bundle statistics and has the objects listed in [Table 177 on page 554](#).

Table 177: jnxL2tpMlpppBundleStatsTable

Object	Object Identifier	Description
jnxL2tpMlpppBundleStatsBundleID	jnxL2tpMlpppBundleStatsEntry 1	Identifies the session's associated bundle.
jnxL2tpMlpppBundleStatsNumLinks	jnxL2tpMlpppBundleStatsEntry 2	Shows the current number of links that have joined the bundle.
jnxL2tpMlpppBundleStatsEndpoint	jnxL2tpMlpppBundleStatsEntry 3	Shows the username of the MLPPP bundle.
jnxL2tpMlpppBundleStatsInputMrru	jnxL2tpMlpppBundleStatsEntry 4	Shows the maximum packet size that the input interface can process.
jnxL2tpMlpppBundleStatsOutputMrru	jnxL2tpMlpppBundleStatsEntry 5	Shows the maximum packet size that the output interface can process.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 43

LDP MIB

- [Interpreting the Enterprise-Specific LDP MIB on page 557](#)

Interpreting the Enterprise-Specific LDP MIB

- [LDP MIB on page 557](#)
- [Juniper Networks Enterprise-Specific LDP Traps on page 558](#)
- [jnxLdpStatsTable on page 560](#)

LDP MIB

The Juniper Networks enterprise-specific LDP MIB, whose object identifier is **{jnxMibs14}**, contains LDP statistics, and defines LDP notification objects and types.

The enterprise-specific LDP MIB uses the following objects and definitions from standard MIBs and enterprise-specific MIB definitions:

- **IpAddress** from SNMPv2-SMI MIB
- **DisplayString** from SNMPv2-TC MIB
- **InterfaceIndex** and **InterfaceIndexOrzero** from IF MIB
- **jnxMibs** and **jnxLdpTraps** from Juniper Enterprise-Specific SMI MIB
- **jnxMplsLdpSesState** from Juniper Enterprise-Specific MPLS LDP MIB
- **MplsVpnName** from the standard MPLS VPN MIB
- **InetAddressType**, **InetAddress**, and **InetAddressPrefixLength** from the standard Inet Address MIB

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ldp.txt.

For information about the enterprise-specific LDP MIB objects, see the following topics:

- [Juniper Networks Enterprise-Specific LDP Traps on page 558](#)
- [jnxLdpStatsTable on page 560](#)

Juniper Networks Enterprise-Specific LDP Traps

For information about the enterprise-specific LDP traps, see “LDP MIB” on page 557.

LDP Notification Objects and Notification Types

The `jnxLdpTrapVars` table, whose object identifier is `{jnxLdp 1}`, defines the objects used in the enterprise-specific LDP traps. For more information about LDP notification types, see Table 178 on page 558.



NOTE: The enterprise-specific LDP MIB requires `jnxLdpTrapPrefix` with 0 subidentifier for seamless translation of SNMPv2 notifications to SNMPv1 format.

Table 178: LDP Notification Objects

Object	Object ID	Description
<code>jnxLdpLspFec</code>	<code>jnxLdpTrapVars 1</code>	Contains the LSP forwarding equivalence class (FEC) in <code>IpAddress</code> format.
<code>jnxLdpRtrid</code>	<code>jnxLdpTrapVars 2</code>	Contains the router ID of the sending router in <code>IpAddress</code> format.
<code>jnxLdpLspDownReason</code>	<code>jnxLdpTrapVars 3</code>	<p>Returns one of the following integer values to denote what might have caused the LSP to go down:</p> <ul style="list-style-type: none"> 1—Change in topology 2—Withdrawal of label by the neighbor 3—Unavailability of the neighbor because the neighbor is down 4—Change in filter 5—Unknown reason
<code>jnxLdpSesDownReason</code>	<code>jnxLdpTrapVars 4</code>	<p>Returns one of the following integer values to denote what might have caused the session transition into nonexistent state:</p> <ul style="list-style-type: none"> 0—Unknown reason 1—Hold time expired 2—Connection time expired 3—All adjacencies are down 4—Received bad TLV (type, length, and value encoding scheme) 5—Received bad PDU (protocol data unit) 6—Connection error occurred 7—The connection was reset 8—Received notification from peer 9—Received unexpected end-of-file message 10—The authentication key was changed 11—Error occurred during initialization 12—Graceful restart was aborted 13—CLI command was issued to end the session

Table 178: LDP Notification Objects (*continued*)

Object	Object ID	Description
<code>jnxLdpSesDownIf</code>	<code>jnxLdpTrapVars 5</code>	Contains the SNMP index of the interface associated with the session-down event. If no interface is associated with the session-down event, then this object returns the SNMP index of any interface associated with one of the neighbors.
<code>jnxLdpLspFecLen</code>	<code>jnxLdpTrapVars 6</code>	Represents the length of the LSP FEC prefix in bits. The allowable range is 0 through 32 bits.
<code>jnxLdpSesUpIf</code>	<code>jnxLdpTrapVars 7</code>	Contains the interface index of one of the neighbors associated with the session.
<code>jnxLdpInstanceName</code>	<code>jnxLdpTrapVars 8</code>	Contains the name of the VPN interface.

Table 179 on page 559 lists the enterprise-specific LDP notifications and the objects contained in each notification. The enterprise-specific LDP notifications use the objects listed in Table 178 on page 558.

Table 179: LDP Notification Types

Object	Object ID	Description
<code>jnxLdpLspUp</code>	<code>jnxLdpTrapPrefix 1</code>	<p>Generated when an LSP comes back online. Typically, this trap is generated only when an LSP that has an active <code>jnxLdpLspDown</code> trap comes back online.</p> <ul style="list-style-type: none"> • <code>jnxLdpLspFec</code> • <code>jnxLdpRtrid</code> • <code>jnxLdpLspFecLen</code> • <code>jnxLdpInstanceName</code>
<code>jnxLdpLspDown</code>	<code>jnxLdpTrapPrefix 2</code>	<p>Generated when an LSP goes offline. This trap contains the following objects:</p> <ul style="list-style-type: none"> • <code>jnxLdpLspFec</code> • <code>jnxLdpRtrid</code> • <code>jnxLdpLspDownReason</code> • <code>jnxLdpLspFecLen</code> • <code>jnxLdpInstanceName</code> <p>NOTE: For every <code>jnxLdpLspDown</code> trap generated, Junos OS ensures that a <code>jnxLdpLspUp</code> trap is generated when the LSP comes back online.</p>

Table 179: LDP Notification Types (*continued*)

Object	Object ID	Description
<code>jnxLdpSesUp</code>	<code>jnxLdpTrapPrefix 3</code>	<p>Generated when the <code>jnxMplsLdpSesState</code> object moves into the operational (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> • <code>jnxMplsLdpSesState</code> • <code>jnxLdpSesUpIf</code>
<code>jnxLdpSesDown</code>	<code>jnxLdpTrapPrefix 4</code>	<p>Generated when the <code>jnxMplsLdpSesState</code> object moves out of the operational (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> • <code>jnxMplsLdpSesState</code> • <code>jnxLdpSesDownReason</code> • <code>jnxLdpSesDownIf</code> <p>The <code>jnxLdpSesDownIf</code> object contains the address of the interface associated with the last neighbor when the value of <code>jnxLdpSesDownReason</code> was <code>allAdjacenciesDown (3)</code>.</p>

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.

`jnxLdpStatsTable`

The `jnxLdpStatsTable`, whose object identifier is `{jnxLdp 2}`, contains the statistics associated with a particular LDP FEC. Each `jnxLdpStatsEntry` in `jnxLdpStatsTable` contains the objects listed in [Table 180 on page 560](#).

Table 180: `jnxLdpStatsTable`

Object	Object ID	Description
<code>jnxLdpInstanceId</code>	<code>jnxLdpStatsEntry 1</code>	Identifies the LDP instance.
<code>jnxLdpFecType</code>	<code>jnxLdpStatsEntry 2</code>	Denotes the type of the LDP instance.
<code>jnxLdpFec</code>	<code>jnxLdpStatsEntry 3</code>	Contains the <code>InetAddress</code> of the LDP FEC.
<code>jnxLdpFecLength</code>	<code>jnxLdpStatsEntry 4</code>	Shows the LDP FEC length in bits. The allowable range is 0 through 32 bits.

Table 180: jnxLdpStatsTable (*continued*)

Object	Object ID	Description
jnxLdpFecStatisticsStatus	jnxLdpStatsEntry 5	<p>Contains one of the following integer values to indicate the status of traffic statistics for the FEC:</p> <ul style="list-style-type: none"> 1—Enabled and available 2—Disabled 3—Unavailable <p>The traffic statistics can be disabled for the penultimate hop FECs, and in such cases, the objects after jnxLdpFecStatisticsStatus in the jnxLdpStatsEntry return 0 value.</p>
jnxLdpIngressOctets	jnxLdpStatsEntry 6	Shows the number of octets of traffic originated from the router, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object might return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
jnxLdpIngressPackets	jnxLdpStatsEntry 7	Shows the number of packets originated from the router, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object might return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
jnxLdpTransitOctets	jnxLdpStatsEntry 8	Shows the number of octets of traffic originated from a different router, but destined for this FEC, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object might return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.
jnxLdpTransitPackets	jnxLdpStatsEntry 9	Shows the number of packets of traffic originated from a different router, but destined for this FEC, and forwarded over the current LDP FEC. Because the LDP statistics are collected at preconfigured intervals and not in real time, this object might return a value that is different from the current value. The default interval for LDP statistics collection is 5 minutes.

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 44

License MIB

- [Interpreting the Enterprise-Specific License MIB on page 563](#)

Interpreting the Enterprise-Specific License MIB

- [License MIB on page 563](#)
- [jnxLicenseInstallTable on page 564](#)
- [Supported Features and Configuration Parameters for Licensing on page 564](#)
- [Juniper Networks Enterprise-Specific License MIB Notifications on page 565](#)

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/junos13.2/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 564](#)
- [Supported Features and Configuration Parameters for Licensing on page 564](#)
- [Juniper Networks Enterprise-Specific License MIB Notifications on page 565](#)

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 181 on page 564](#).

Table 181: 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.

Supported Features and Configuration Parameters for Licensing

This topic contains the following sections:

- [jnxLicenseFeatureListTable on page 564](#)
- [License Configuration Parameters on page 565](#)

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 182 on page 564](#).

Table 182: jnxLicenseFeatureListEntry

Object	Object Identifier	Description
jnxLicenseFeatureId	jnxLicenseFeatureListEntry 1	Contains the feature ID that is the index of the jnxLicenseFeatureListEntry .

Table 182: `jnxLicenseFeatureListEntry` (*continued*)

Object	Object Identifier	Description
<code>jnxLicenseFeatureName</code>	<code>jnxLicenseFeatureListEntry 2</code>	Contains the name of the feature.
<code>jnxLicenseFeatureDescr</code>	<code>jnxLicenseFeatureListEntry 3</code>	Contains a description of the feature.
<code>jnxLicenseFeatureLicenseld</code>	<code>jnxLicenseFeatureListEntry 4</code>	Contains the feature license ID (from the <code>jnxLicenseInstallTable</code>) 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:

- `jnxLicenseRenewBeforeExpiration`, 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.
- `jnxLicenseAutoUpdate`, 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).

Juniper Networks Enterprise-Specific License MIB Notifications

Table 183 on page 565 lists the license MIB notifications.

Table 183: License MIB Notifications

Notification	Object Identifier	Description
<code>jnxLicenseGraceExpired</code>	<code>jnxLicenseNotifications 1</code>	Generated when the grace period for renewing the license for the feature (identified by <code>jnxLicenseFeatureName</code>) has expired.
<code>jnxLicenseGraceAboutToExpire</code>	<code>jnxLicenseNotifications 2</code>	Generated when the grace period for renewing the license for the feature (identified by <code>jnxLicenseFeatureName</code>) is about to expire.
<code>jnxLicenseAboutToExpire</code>	<code>jnxLicenseNotifications 3</code>	Generated when the license for the feature (identified by <code>jnxLicenseFeatureName</code>) is about to expire.
<code>jnxLicenseInfringeCumulative</code>	<code>jnxLicenseNotifications 4</code>	Generated when the use of the feature (identified by <code>jnxLicenseFeatureName</code>) exceeds the maximum limit specified in the license.

Table 183: License MIB Notifications (*continued*)

Notification	Object Identifier	Description
jnxLicenseInfringeSingle	jnxLicenseNotifications 5	Generated when the license for the feature (identified by jnxLicenseFeatureName) is invalid; that is, either expired or not available.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 45

Logical Systems MIBs

- [Interpreting the Enterprise-Specific Logical Systems \(LSYS\) MIBs on page 567](#)

Interpreting the Enterprise-Specific Logical Systems (LSYS) MIBs

- [LSYS Security Profile on page 567](#)
- [LSYS Zone MIB on page 568](#)
- [LSYS Scheduler MIB on page 570](#)
- [LSYS Policy MIB on page 572](#)
- [LSYS Policy with Count MIB on page 574](#)
- [LSYS Flow Gate MIB on page 576](#)
- [LSYS Flow Session MIB on page 578](#)
- [LSYS AUTH Entry MIB on page 580](#)
- [LSYS NAT Source Pool MIB on page 582](#)
- [LSYS NAT Destination Pool MIB on page 584](#)
- [LSYS NAT Source PAT Address MIB on page 586](#)
- [LSYS NAT Source No PAT Address MIB on page 588](#)
- [LSYS NAT Source Rule MIB on page 590](#)
- [LSYS NAT Destination Rule MIB on page 592](#)
- [LSYS NAT Static Rule MIB on page 595](#)
- [LSYS NAT Cone Bind MIB on page 597](#)
- [LSYS NAT Port Overloading IP Number MIB on page 599](#)
- [LSYS CPU MIB on page 601](#)

LSYS Security Profile

- [jnxLsysSecurityProfile on page 567](#)

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/junos13.2/topics/
reference/mibs/mib-jnx-lsys-securityprofile.txt](http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-lsys-securityprofile.txt).

For information about the logical system security profile, see the following topics:

- [jnxLsysSpZoneMIB on page 568](#)
- [jnxLsysSpSchedulerMIB on page 570](#)
- [jnxLsysSpPolicyMIB on page 572](#)
- [jnxLsysSpPolicywcntSummary on page 575](#)
- [jnxLsysSpFlowgateMIB on page 577](#)
- [jnxLsysSpFlowsessMIB on page 578](#)
- [jnxLsysSpAuthentryMIB on page 580](#)
- [jnxLsysSpNATsrcpoolMIB on page 582](#)
- [jnxLsysSpNATdstpoolMIB on page 584](#)
- [jnxLsysSpNATsrcpatadMIB on page 586](#)
- [jnxLsysSpNATsrcnopatadMIB on page 588](#)
- [jnxLsysSpNATsrcruleMIB on page 590](#)
- [jnxLsysSpNATdstruleMIB on page 593](#)
- [jnxLsysSpNATstaticruleMIB on page 595](#)
- [jnxLsysSpNATconebindMIB on page 597](#)
- [jnxLsysSpNATpoipnumMIB on page 599](#)
- [jnxLsysSpCPUMIB on page 601](#)

LSYS Zone MIB

- [jnxLsysSpZoneMIB on page 568](#)
- [jnxLsysSpZoneObjects on page 569](#)
- [jnxLsysSpZoneSummary on page 569](#)

jnxLsysSpZoneMIB

The Juniper Networks enterprise-specific zone MIB, whose object identifier is **{jnxLsysSpZone 1 }**, defines the zone-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see
[http://www.juniper.net/techpubs/en_US/junos13.2/topics/
reference/mibs/mib-jnx-lsys-sp-zone.txt](http://www.juniper.net/techpubs/en_US/junos13.2/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 569](#)

- [jnxLsysSpZoneSummary](#) on page 569

[jnxLsysSpZoneObjects](#)

The **jnxLsysSpZoneObjects**, whose object identifier is **{jnxLsysSpZoneMIB 1}**, includes the logical system zone-specific objects as listed in [Table 184 on page 569](#).

Table 184: 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 185 on page 569](#) lists and describes the objects contained in a **jnxLsysSpZoneEntry**.

Table 185: 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.

[jnxLsysSpZoneSummary](#)

The **jnxLsysSpZoneSummary**, whose object identifier is **{jnxLsysSpZoneMIB 2 }**, represents the zone resource consumption summary.

[Table 186 on page 569](#) lists and describes the zone resource consumption summary.

Table 186: jnxLsysSpZoneSummary

Object	Object ID	Description
jnxLsysSpZoneUsedAmount	jnxLsysSpZoneSummary 1	Represents the total zone resource consumption by the logical systems.

Table 186: jnxLsysSpZoneSummary (continued)

Object	Object ID	Description
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.

LSYS Scheduler MIB

- jnxLsysSpSchedulerMIB on page 570
- jnxLsysSpSchedulerObjects on page 571
- jnxLsysSpSchedulerSummary on page 571

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/junos13.2/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 571
- jnxLsysSpSchedulerSummary on page 571

jnxLsysSpSchedulerObjects

The **jnxLsysSpSchedulerObjects**, whose object identifier is **{jnxLsysSpSchedulerMIB 1}**, includes the logical system scheduler-specific objects as listed in [Table 187 on page 571](#):

Table 187: 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 188 on page 571](#) lists and describes the objects contained in a **jnxLsysSpSchedulerEntry**.

Table 188: 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.
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.

jnxLsysSpSchedulerSummary

The **jnxLsysSpSchedulerSummary**, whose object identifier is **{jnxLsysSpSchedulerMIB 2}**, represents the scheduler resource consumption summary.

[Table 189 on page 572](#) lists and describes the scheduler resource consumption summary.

Table 189: 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.
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.

LSYS Policy MIB

- [jnxLsysSpPolicyMIB on page 572](#)
- [jnxLsysSpPolicyObjects on page 573](#)
- [jnxLsysSpPolicySummary on page 573](#)

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/junos13.2/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 573](#)
- [jnxLsysSpPolicySummary on page 573](#)

[jnxLsysSpPolicyObjects](#)

The **jnxLsysSpPolicyObjects**, whose object identifier is **{ jnxLsysSpPolicyMIB 1}**, represents the logical system policy objects as listed in [Table 190 on page 573](#).

Table 190: jnxLsysSpZoneObjects

Object	Object Identifier	Description
jnxLsysSpPolicyTable	jnxLsysSpPolicyObjects 1	Represents the logical system profile for policy resource consumption per logical system.
jnxLsysSpPolicyEntry	jnxLsysSpPolicyTable 1	Identifies an entry in the policy resource table.

[Table 191 on page 573](#) lists and describes the objects contained in a **{jnxLsysSpPolicyEntry}**.

Table 191: 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.

[jnxLsysSpPolicySummary](#)

The **jnxLsysSpPolicySummary**, whose object identifier is **{jnxLsysSpPolicyMIB 2}**, represents the policy resource consumption summary.

[Table 192 on page 574](#) lists and describes the policy resource consumption summary.

Table 192: 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.

LSYS Policy with Count MIB

- [jnxLsysSpPolicywcntMIB on page 574](#)
- [jnxLsysSpPolicywcntObjects on page 575](#)
- [jnxLsysSpPolicywcntSummary on page 575](#)

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/junos13.2/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 575](#)

- [jnxLsysSpPolicywcntSummary](#) on page 575

[jnxLsysSpPolicywcntObjects](#)

The **jnxLsysSpPolicywcntObjects**, whose object identifier is **{jnxLsysSpPolicywcntMIB 1}**, represents the logical systems policy-with-count specific objects as listed in [Table 193](#) on page 575.

Table 193: 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 194](#) on page 575 lists and describes the objects contained in a **JnxLsysSpPolicywcntEntry**.

Table 194: 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.
jnxLsysSpPolicywcntMaximum	jnxLsysSpPolicywcntEntry 5	Represents the maximum allowed resource usage count for the logical system.

[jnxLsysSpPolicywcntSummary](#)

The **jnxLsysSpPolicywcntSummary**, whose object identifier is **{jnxLsysSpPolicywcntMIB 2}**, represents the policy-with-count resource consumption summary.

Table 195 on page 576 lists and describes the objects for policy-with-count resource consumption summary.

Table 195: 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.
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.

LSYS Flow Gate MIB

- [jnxLsysSpFlowgateMIB on page 577](#)
- [jnxLsysSpFlowgateObjects on page 577](#)
- [jnxLsysSpFlowgateSummary on page 578](#)

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 [Junos OS System Basics and Services Command Reference](#)

For information about the enterprise-specific logical system NAT flow- gate-specific MIB objects, see the following topics:

- [jnxLsysSpFlowgateObjects](#) on page 577
- [jnxLsysSpFlowgateSummary](#) on page 578

jnxLsysSpFlowgateObjects

The **jnxLsysSpFlowgateObjects**, whose object identifier is **{jnxLsysSpFlowgateMIB 1}**, represents the logical system flow- gate-specific objects as listed in [Table 196 on page 577](#).

Table 196: 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 197 on page 577](#).

Table 197: 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.

[jnxLsysSpFlowgateSummary](#)

The **jnxLsysSpFlowgateSummary**, whose object identifier is **{jnxLsysSpFlowgateMIB 2}**, represents the flow gate resource consumption summary.

[Table 198 on page 578](#) lists and describes the flow-gate resource consumption summary.

Table 198: 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.
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.

LSYS Flow Session MIB

- [jnxLsysSpFlowsessMIB on page 578](#)
- [jnxLsysSpFlowsessObjects on page 579](#)
- [jnxLsysSpFlowsessSummary on page 579](#)

[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 [Junos OS System Basics and Services Command Reference](#)

For information about the enterprise-specific logical system flow session MIB objects, see the following topics:

- [jnxLsysSpFlowsessObjects on page 579](#)
- [jnxLsysSpFlowsessSummary on page 579](#)

[jnxLsysSpFlowsessObjects](#)

The **jnxLsysSpFlowsessObjects**, whose object identifier is **{jnxLsysSpFlowsessMIB 1}**, represents the logical system flow session specific objects as listed in [Table 199 on page 579](#).

Table 199: 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 200 on page 579](#).

Table 200: 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.

[jnxLsysSpFlowsessSummary](#)

The **jnxLsysSpFlowsessSummary**, whose object identifier is **{jnxLsysSpFlowsessMIB 2}**, represents the flow session resource consumption summary.

[Table 201 on page 580](#) lists and describes the flow session resource consumption summary.

Table 201: 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.

LSYS AUTH Entry MIB

- [jnxLsysSpAuthentryMIB on page 580](#)
- [jnxLsysSpAuthentryObjects on page 581](#)
- [jnxLsysSpAuthentrySummary on page 581](#)

jnxLsysSpAuthentryMIB

The Juniper Networks enterprise-specific LSYS NAT Auth entry MIB, whose object identifier is **{jnxLsysSpAuthentry1}**, defines the authentication-entry-specific MIB for logical system security profiles.

For a downloadable version of this MIB, see [Junos OS System Basics and Services Command Reference](#)

For information about the enterprise-specific logical system auth entry pool MIB objects, see the following topics:

- [jnxLsysSpAuthentryObjects on page 581](#)
- [jnxLsysSpAuthentrySummary on page 581](#)

[jnxLsysSpAuthentryObjects](#)

The **jnxLsysSpAuthentryObjects**, whose object identifier is **{jnxLsysSpAuthentryMIB 1}**, represents the logical system NAT authentication entry objects as listed in [Table 202 on page 581](#).

Table 202: 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 203 on page 581](#).

Table 203: 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.

[jnxLsysSpAuthentrySummary](#)

The **jnxLsysSpAuthentrySummary**, whose object identifier is **{jnxLsysSpAuthentryMIB 2}**, represents the authentication entry resource consumption summary.

[Table 204 on page 581](#) lists and describes the authentication entry resource consumption summary.

Table 204: jnxLsysSpAuthentrySummary

Object	Object ID	Description
jnxLsysSpAuthentryUsedAmount	jnxLsysSpAuthentrySummary 1	Represents the authentication entry resource consumption for all logical systems.

Table 204: jnxLsysSpAuthentrySummary (*continued*)

Object	Object ID	Description
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.

LSYS NAT Source Pool MIB

- [jnxLsysSpNATsrcpoolMIB on page 582](#)
- [jnxLsysSpNATsrcpoolObjects on page 583](#)
- [jnxLsysSpNATsrcpoolSummary on page 583](#)

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.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/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 583](#)
- [jnxLsysSpNATsrcpoolSummary on page 583](#)

jnxLsysSpNATsrcpoolObjects

The **jnxLsysSpNATsrcpoolObjects**, whose object identifier is **{jnxLsysSpNATsrcpoolMIB 1}**, represents the logical system NAT source pool objects as listed in [Table 205 on page 583](#).

Table 205: 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 206 on page 583](#).

Table 206: 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.

jnxLsysSpNATsrcpoolSummary

The **jnxLsysSpNATsrcpoolSummary**, whose object identifier is **{jnxLsysSpNATsrcpoolMIB 2}**, represents the NAT source pool resource consumption summary.

[Table 207 on page 583](#) lists and describes the NAT source pool resource consumption summary.

Table 207: jnxLsysSpNATsrcpoolSummary

Object	Object ID	Description
jnxLsysSpNATsrcpoolUsedAmount	jnxLsysSpNATsrcpoolSummary 1	Represents the NAT source pool resource consumption for all logical systems.

Table 207: jnxLsysSpNATsrcpoolSummary (continued)

Object	Object ID	Description
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.

LSYS NAT Destination Pool MIB

- [jnxLsysSpNATdstpoolMIB on page 584](#)
- [jnxLsysSpNATdstpoolObjects on page 585](#)
- [jnxLsysSpNATdstpoolSummary on page 585](#)

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/junos13.2/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 585](#)
- [jnxLsysSpNATdstpoolSummary on page 585](#)

jnxLsysSpNATdstpoolObjects

The **jnxLsysSpNATdstpoolObjects**, whose object identifier is **{jnxLsysSpNATdstpoolMIB 1}**, as listed in [Table 208 on page 585](#).

Table 208: 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 209 on page 585](#).

Table 209: 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.
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.

jnxLsysSpNATdstpoolSummary

The **jnxLsysSpNATdstpoolSummary**, whose object identifier is **{jnxLsysSpNATdstpoolMIB 2}**, represents the NAT destination pool resource consumption summary.

[Table 210 on page 586](#) lists and describes the NAT destination pool resource consumption summary.

Table 210: 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.

LSYS NAT Source PAT Address MIB

- [jnxLsysSpNATsrcpatadMIB on page 586](#)
- [jnxLsysSpNATsrcpatadObjects on page 587](#)
- [jnxLsysSpNATsrcpatadSummary on page 587](#)

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/junos13.2/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 587](#)
- [jnxLsysSpNATsrcpatadSummary on page 587](#)

[jnxLsysSpNATsrcpatadObjects](#)

The **jnxLsysSpNATsrcpatadObjects**, whose object identifier is **{jnxLsysSpNATsrcpatadMIB 1}**, represents the logical system NAT source PAT address objects as listed in [Table 211 on page 587](#).

Table 211: 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 212 on page 587](#).

Table 212: 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.

[jnxLsysSpNATsrcpatadSummary](#)

The **jnxLsysSpNATsrcpatadSummary**, whose object identifier is **{jnxLsysSpNATsrcpatadMIB 2}**, represents the logical system NAT source PAT address resource consumption summary.

[Table 213 on page 588](#) lists and describes NAT source PAT address resource consumption summary.

Table 213: 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.
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.

LSYS NAT Source No PAT Address MIB

- [jnxLsysSpNATsrcnopatadMIB on page 588](#)
- [jnxLsysSpNATsrcnopatadObjects on page 589](#)
- [jnxLsysSpNATsrcnopatadSummary on page 589](#)

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/junos13.2/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 589](#)
- [jnxLsysSpNATsrcnopatadSummary on page 589](#)

jnxLsysSpNATsrcnopatadObjects

The **jnxLsysSpNATsrcnopatadObjects**, whose object identifier is **{jnxLsysSpNATsrcnopatadMIB 1}**, represents the logical system NAT source no PAT address objects as listed in [Table 214 on page 589](#).

Table 214: 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 215 on page 589](#).

Table 215: 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.

jnxLsysSpNATsrcnopatadSummary

The **jnxLsysSpNATsrcnopatadSummary**, whose object identifier is **{jnxLsysSpNATsrcnopatadMIB 2}**, represents the NAT source no PAT address resource consumption summary.

[Table 216 on page 590](#) lists and describes the NAT source pool resource consumption summary.

Table 216: 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.

LSYS NAT Source Rule MIB

- [jnxLsysSpNATsrcruleMIB on page 590](#)
- [jnxLsysSpNATsrcruleObjects on page 591](#)
- [jnxLsysSpNATsrcruleSummary on page 591](#)

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/junos13.2/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 591](#)
- [jnxLsysSpNATsrcruleSummary on page 591](#)

[jnxLsysSpNATsrcruleObjects](#)

The **jnxLsysSpNATsrcruleObjects**, whose object identifier is { **jnxLsysSpNATsrcruleMIB 1** }, represents the logical system NAT source rule objects as listed in [Table 217 on page 591](#).

Table 217: 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 218 on page 591](#).

Table 218: 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.
jnxLsysSpNATsrcruleMaximum	jnxLsysSpNATsrcruleEntry 5	Represents the maximum allowed resource usage count for the logical system.

[jnxLsysSpNATsrcruleSummary](#)

The **jnxLsysSpNATsrcruleSummary** includes the objects that are listed in [Table 219 on page 592](#).

Table 219: 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.
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.

LSYS NAT Destination Rule MIB

- [jnxLsysSpNATdsteruleMIB on page 593](#)
- [jnxLsysSpNATdsteruleObjects on page 593](#)
- [jnxLsysSpNATdsteruleSummary on page 594](#)

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/junos13.2/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 593](#)
- [jnxLsysSpNATdstruleSummary on page 594](#)

jnxLsysSpNATdstruleObjects

The **jnxLsysSpNATdstruleObjects**, whose object identifier is **{jnxLsysSpNATdstruleMIB 1}**, represents the logical system NAT destination rule objects as listed in [Table 220 on page 593](#).

Table 220: 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 221 on page 593](#).

Table 221: 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.

Table 221: jnxLsysSpNATdstRuleEntry (*continued*)

Object	Object ID	Description
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.

jnxLsysSpNATdstRuleSummary

The **jnxLsysSpNATdstRuleSummary**, whose object identifier is **{jnxLsysSpNATdstRuleMIB 2}**, represents the NAT destination rule resource consumption summary.

[Table 222 on page 594](#) lists and describes the NAT destination rule resource consumption summary.

Table 222: 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.

LSYS NAT Static Rule MIB

- [jnxLsysSpNATstaticruleMIB on page 595](#)
- [jnxLsysSpNATstaticruleObjects on page 595](#)
- [jnxLsysSpNATstaticruleSummary on page 596](#)

[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/junos13.2/topics/reference/mibs/mib-jnx-lsys-sp-natstaticrule.txt.

For information about the enterprise-specific logical system NAT static rule-specific MIB objects, see the following topics:

- [jnxLsysSpNATstaticruleObjects on page 595](#)
- [jnxLsysSpNATstaticruleSummary on page 596](#)

[jnxLsysSpNATstaticruleObjects](#)

The **jnxLsysSpNATstaticruleObjects**, whose object identifier is {**jnxLsysSpNATstaticruleMIB 1**}, represents the logical system NAT static rule objects as listed in [Table 223 on page 595](#).

Table 223: 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 224 on page 595](#).

Table 224: jnxLsysSpNATstaticruleEntry

Object	Object ID	Description
jnxLsysSpNATstaticruleLsysName	jnxLsysSpNATstaticruleEntry 1	Represents the name of the logical system that uses the NAT static rule resource information.

Table 224: jnxLsysSpNATstaticruleEntry (*continued*)

Object	Object ID	Description
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.
jnxLsysSpNATstaticruleMaximum	jnxLsysSpNATstaticruleEntry 5	Represents the maximum allowed resource usage count for the logical system.

jnxLsysSpNATstaticruleSummary

The `jnxLsysSpNATstaticruleSummary`, whose object identifier is `{jnxLsysSpNATstaticruleMIB 2 }`, defines the objects that are used to monitor NAT attributes as listed in [Table 225 on page 596](#).

Table 225: jnxLsysSpNATstaticruleSummary

Object	Object ID	Description
jnxLsysSpNATstaticruleUsedAmount	jnxLsysSpNATstaticruleSummary 1	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.

Table 225: jnxLsysSpNATstaticruleSummary (*continued*)

Object	Object ID	Description
jnxLsysSpNATstaticruleLighttestUsage	jnxLsysSpNATstaticruleSummary 6	Represents the minimum NAT static rule resource consumption by a logical system.
jnxlogical systemsSpNATsrcruleLighttestUser	jnxLsysSpNATstaticruleSummary 7	Represents the name of the logical system that consumes the minimum NAT static rule resources.

LSYS NAT Cone Bind MIB

- [jnxLsysSpNATconebindMIB on page 597](#)
- [jnxLsysSpNATconebindObjects on page 597](#)
- [jnxLsysSpNATconebindSummary on page 598](#)

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 [Junos OS System Basics and Services Command Reference](#)

For information about the enterprise-specific logical system NAT cone bind MIB objects, see the following topics:

- [jnxLsysSpNATconebindObjects on page 597](#)
- [jnxLsysSpNATconebindSummary on page 598](#)

jnxLsysSpNATconebindObjects

The **jnxLsysSpNATconebindObjects**, whose object identifier is **{jnxLsysSpNATconebindMIB 1}**, represents the logical system NAT cone bind specific objects as listed in [Table 226 on page 597](#).

Table 226: 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.
jnxLsysSpNATconebindEntry	jnxLsysSpNATconebindTable 1	Represents an entry in the NAT cone bind resource table.

The **JnxLsysSpNATconebindEntry** includes the objects that are listed in [Table 227 on page 598](#).

Table 227: 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.

jnxLsysSpNATconebindSummary

The **jnxLsysSpNATconebindSummary**, whose object identifier is {jnxLsysSpNATconebindMIB 2}, represents the NAT cone bind resource consumption summary.

[Table 228 on page 598](#) lists and describes the NAT cone bind resource consumption summary.

Table 228: 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.
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.

Table 228: jnxLsysSpNATconebindSummary (*continued*)

Object	Object ID	Description
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.

LSYS NAT Port Overloading IP Number MIB

- [jnxLsysSpNATpoipnumMIB on page 599](#)
- [jnxLsysSpNATpoipnumObjects on page 599](#)
- [jnxLsysSpNATpoipnumSummary on page 600](#)

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/junos13.2/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 599](#)
- [jnxLsysSpNATpoipnumSummary on page 600](#)

jnxLsysSpNATpoipnumObjects

The **jnxLsysSpNATpoipnumObjects**, whose object identifier is **{jnxLsysSpNATpoipnumMIB 1}**, represents the logical system NAT port overloading IP number objects as listed in [Table 229 on page 599](#).

Table 229: 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 230 on page 600](#).

Table 230: 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.

jnxLsysSpNATpoipnumSummary

The **jnxLsysSpNATpoipnumSummary**, whose object identifier is {**jnxLsysSpNATpoipnumMIB 2**}, represents the NAT port overloading IP number resource consumption summary.

[Table 231 on page 600](#) lists and describes the NAT source pool resource consumption summary.

Table 231: 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.

Table 231: jnxLsysSpNATpoipnumSummary (*continued*)

Object	Object ID	Description
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.

LSYS CPU MIB

- [jnxLsysSpCPUMIB on page 601](#)
- [jnxLsysSpCPUObjects on page 601](#)
- [jnxLsysSpCPSummary on page 602](#)
- [jnxLsysSpSPUSummary on page 603](#)

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 [Junos OS System Basics and Services Command Reference](#)

For information about the enterprise-specific logical system CPU MIB objects, see the following topics:

- [jnxLsysSpCPUObjects on page 601](#)
- [jnxLsysSpCPSummary on page 602](#)
- [jnxLsysSpSPUSummary on page 603](#)

jnxLsysSpCPUObjects

The **jnxLsysSpCPUObjects**, whose object identifier is **{jnxLsysSpCPUMIB 1}**, represents the logical system CPU objects as listed in [Table 232 on page 601](#).

Table 232: 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 233 on page 602](#).

Table 233: 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.
jnxLsysSpCPUUsage	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.

jnxLsysSpCPSummary

The **jnxLsysSpCPSummary**, whose object identifier is **{jnxLsysSpCPSummary 1}**, represents the CP resource consumption summary.

Table 234 on page 602 lists and describes the CP resource consumption summary.

Table 234: 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.

[jnxLsysSpSPUSummary](#)

The `jnxLsysSpSPUSummary`, whose object identifier is `{jnxLsysSpCPUSummary 2}`, represents the SPU resource consumption summary.

Table 235 on page 603 lists and describes the SPU resource consumption summary.

Table 235: jnxLsysSpSPUSummary

Object	Object ID	Description
<code>jnxLsysSpSPUUsedAmount</code>	<code>jnxLsysSpSPUSummary 1</code>	Represents the SPU resource consumption for all logical systems.
<code>jnxLsysSpSPUMaxQuota</code>	<code>jnxLsysSpSPUSummary 2</code>	Represents the SPU resource maximum quota for all logical systems on the device.
<code>jnxLsysSpSPUAvailableAmount</code>	<code>jnxLsysSpSPUSummary 3</code>	Represents the amount of SPU resources available for all logical systems on the device.
<code>jnxLsysSpSPUHeaviestUsage</code>	<code>jnxLsysSpSPUSummary 4</code>	Represents the most used SPU resources consumed by a logical system.
<code>jnxLsysSpSPUHeaviestUser</code>	<code>jnxLsysSpSPUSummary 5</code>	Represents the logical system name that consumes the most SPU resources.
<code>jnxLsysSpSPULightestUsage</code>	<code>jnxLsysSpSPUSummary 6</code>	Represents the least used SPU resources consumed by a logical system.
<code>jnxLsysSpSPULightestUser</code>	<code>jnxLsysSpSPUSummary 7</code>	Represents the logical system name that consumes the least SPU resources.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 46

MIMSTP MIB

- Interpreting the Enterprise-Specific MIMSTP MIB on page 605

Interpreting the Enterprise-Specific MIMSTP MIB

- MIMSTP MIB on page 605
- jnxMIDot1sJuniperMstTable on page 606
- jnxMIMstMstiBridgeTable on page 612
- jnxMIMstVlanInstanceMappingTable on page 614
- jnxMIMstCistPortTable on page 615
- jnxMIMstMstiPortTable on page 618
- Juniper Networks Enterprise-Specific MIMSTP Traps on page 620

MIMSTP MIB

The Junos OS provides SNMP support for spanning-tree protocols on MX Series 3D Universal Edge Routers .

The following standard and Juniper Networks enterprise-specific MIBs have been added to extend SNMP support to spanning-tree protocols:

- RFC 4188, *Definitions of Managed Objects for Bridges*—Supports 802.1d STP (1998) only.
- RFC 4318, *Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol*—Supports 802.1w and 802.1t extensions for RSTP.
- Juniper Networks Enterprise-Specific Multiple Instance Virtual Switch MIB (**mib-jnx-mimstp.txt**)—Supports 802.1s (2002) for MSTP.

The Juniper Networks enterprise-specific Multiple Instance Multiple Spanning Tree Protocol (MIMSTP) MIB (**mib-jnx-mimstp.txt**) provides information about multiple spanning-tree instances, that is, routing instances of type Virtual Switch/Layer 2 control, also known as virtual contexts and associated VLANs.

For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mimstp.txt .

For information about the enterprise-specific MIMSTP MIB objects, see the following topics:

- [jnxMIDot1sJuniperMstTable](#) on page 606
- [jnxMIMstMstiBridgeTable](#) on page 612
- [jnxMIMstVlanInstanceMappingTable](#) on page 614
- [jnxMIMstCistPortTable](#) on page 615
- [jnxMIMstMstiPortTable](#) on page 618
- [Juniper Networks Enterprise-Specific MIMSTP Traps](#) on page 620

jnxMIDot1sJuniperMstTable

The **jnxMIDot1sJuniperMstTable** provides MSTP module parameters for a given virtual context. [Table 236 on page 606](#) lists the objects in the **jnxMIDot1sJuniperMstTable**.

Table 236: jnxMIDot1sJuniperMstTable

Object	Object Identifier	Description
jnxMIDot1sJuniperMstContextId	jnxMIDot1sJuniperMstEntry 1	Identifies the virtual context.
jnxMIMstSystemControl	jnxMIDot1sJuniperMstEntry 2	Indicates the status of MST on the ports of the device. The value start (1) indicates that MIMST is active on all ports of the device; the value shutdown (2) indicates that MIMST is shut down on all ports of the device.
jnxMIMstModuleStatus	jnxMIDot1sJuniperMstEntry 3	Indicates whether MST is enabled or disabled on the ports. When set to value 1, enabled , MST is enabled on all ports of the device; when set to value 2, disabled , MIMST is disabled on all ports. NOTE: The object can be set to enabled(1) only when jnxMIMstSystemControl is set to start.
jnxMIMstMaxMstInstanceNumber	jnxMIDot1sJuniperMstEntry 4	Indicates the maximum number of spanning-tree instances allowed on the bridge.
jnxMIMstNoOfMstiSupported	jnxMIDot1sJuniperMstEntry 5	Indicates the maximum number of spanning-tree instances that can be supported on the bridge.
jnxMIMstMaxHopCount	jnxMIDot1sJuniperMstEntry 6	Returns the Maximum Hop Count value.

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstBrgAddress	jnxMIDot1sJuniperMstEntry 7	Indicates the MAC address used by the bridge, which forms a unique bridge identifier when combined with jnxMIMstCistBridgePriority or jnxMIMstMstiBridgePriority .
jnxMIMstCistRoot	jnxMIDot1sJuniperMstEntry 8	Indicates the bridge identifier of the root of the common spanning-tree as determined by the Spanning Tree Protocol. This value is used as the common and internal spanning tree (CIST) Root Identifier parameter in all configuration bridge PDUs originating at this node.
jnxMIMstCistRegionalRoot	jnxMIDot1sJuniperMstEntry 9	Indicates the bridge identifier of the root of the multiple spanning-tree region as determined by the Spanning Tree Protocol. This value is used as the CIST Regional Root Identifier parameter in all configuration bridge PDUs originating at this node.
jnxMIMstCistRootCost	jnxMIDot1sJuniperMstEntry 10	Indicates the cost of the path to the CIST root from this bridge.
jnxMIMstCistRegionalRootCost	jnxMIDot1sJuniperMstEntry 11	Indicates the cost of the path to the CIST regional root from this bridge.
jnxMIMstCistRootPort	jnxMIDot1sJuniperMstEntry 12	Indicates the port number of the port that offers the lowest path cost from this bridge to the CIST Root Bridge.
jnxMIMstCistBridgePriority	jnxMIDot1sJuniperMstEntry 13	Indicates the value of the writable portion of the bridge identifier. The values set for Bridge Priority must be in multiples of 4096.
jnxMIMstCistBridgeMaxAge	jnxMIDot1sJuniperMstEntry 14	Indicates the value that a bridge uses for MaxAge when the bridge is acting as the root.
jnxMIMstCistBridgeForwardDelay	jnxMIDot1sJuniperMstEntry 15	Indicates the value that the bridge uses for ForwardDelay when this bridge is acting as the root. Note that 802.1D specifies that the range for this parameter is related to the value of BridgeMaxAge .
jnxMIMstCistHoldTime	jnxMIDot1sJuniperMstEntry 16	Sets the interval between transmitting two configuration bridge PDUs.

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstCistMaxAge	jnxMIDot1sJuniperMstEntry 17	Sets the maximum age of Spanning Tree Protocol (STP) information learned on the ports. The STP information is discarded when the age exceeds the set limit.
jnxMIMstCistForwardDelay	jnxMIDot1sJuniperMstEntry 18	Indicates the time period during which a port stays in a particular state before moving to the next state; for example, from spanning to forwarding state.
jnxMIMstMstpUpCount	jnxMIDot1sJuniperMstEntry 19	Shows the number of times the MSTP Module has been enabled on the bridge.
jnxMIMstMstpDownCount	jnxMIDot1sJuniperMstEntry 20	Shows the number of times the MSTP Module has been disabled on the bridge.
jnxMIMstPathCostDefaultType	jnxMIDot1sJuniperMstEntry 21	Shows the version of the spanning-tree default path costs that are to be used by the bridge. <ul style="list-style-type: none"> • A value of 8021d1998(1) uses the 16-bit default path costs from IEEE Std. 802.1D-1998. • A value of stp8021t2001(2) uses the 32-bit default path costs from IEEE Std. 802.1t.

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstTrace	jnxMIDot1sJuniperMstEntry 22	<p>Enables trace statements in the MSTP module. A 4-byte integer can be used to represent the level of trace.</p> <p>The bit position maps to the following levels of trace:</p> <ul style="list-style-type: none"> • 0: Init and shutdown traces • 1: Management traces • 2: Data path traces • 3: Control plane traces • 4: Packet dump traces • 5: Traces that are related to all resources except buffers • 6: All failure traces • 7: Buffer traces <p>The rest of the bits remain unused. You can use a combination of trace levels to generate trace statements of multiple trace levels.</p> <p>NOTE: Trace options and debug options are mutually exclusive. When the trace option is set, the debug option is set to 0.</p>

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstDebug	jnxMIDot1sJuniperMstEntry 23	<p>Enables debug statements in the MSTP module. A 4-byte integer can be used to set the level of debugging.</p> <p>The bit position maps to the following levels of debugging:</p> <ul style="list-style-type: none"> • 0: Init and shutdown debug statements • 1: Management debug statements • 2: Memory-related debug statements • 3: BPDU-related debug statements • 4: Event handling debug statements • 5: Time module debug statements • 6: Port information SEM debug statements • 7: Port receive SEM debug statements (valid in the case of MSTP alone) • 8: Role selection SEM debug statements • 9: Role transition SEM debug statements • 10: State transition SEM debug statements • 11: Protocol migration SEM debug statements • 12: Topology change SEM debug statements • 13: Port transmit SEM debug statements • 14: Bridge detection SEM debug statements • 15: All failure debug statements • 16: Redundancy code flow debug statements <p>The rest of the bits remain unused. You can use a combination of debug levels to generate debug statements of multiple debug levels.</p> <p>NOTE: Debug options and trace options are mutually exclusive. When the debug option is set, the trace option is set to 0.</p>

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstForceProtocolVersion	jnxMIDot1sJuniperMstEntry 24	<p>Indicates the version of the Spanning Tree Protocol that is running on the bridge.</p> <ul style="list-style-type: none"> • stpCompatible(0): Represents the Spanning Tree Protocol specified in IEEE 802.1D. • rstp(2): Represents the Rapid Spanning Tree Protocol specified in IEEE 802.1w. • mstp(3): Represents the Multiple Spanning Tree Protocol specified in IEEE 802.1s.
jnxMIMstTxHoldCount	jnxMIDot1sJuniperMstEntry 25	Indicates the value that the port transmit state machine uses to limit the maximum transmission rate.
jnxMIMstMstiConfigIdSel	jnxMIDot1sJuniperMstEntry 26	Indicates the Configuration Identifier Format Selector that the bridge uses.
jnxMIMstMstiRegionName	jnxMIDot1sJuniperMstEntry 27	Indicates the name of the region's configuration. By default, the region name and the MAC address of the bridge are the same.
jnxMIMstMstiRegionVersion	jnxMIDot1sJuniperMstEntry 28	Indicates the version of the multiple spanning-tree region.
jnxMIMstMstiConfigDigest	jnxMIDot1sJuniperMstEntry 29	Indicates the configuration digest value for the multiple spanning-tree region.
jnxMIMstBufferOverflowCount	jnxMIDot1sJuniperMstEntry 30	Indicates the number of times buffer overflows or failures have occurred. This event generates a trap.
jnxMIMstMemAllocFailureCount	jnxMIDot1sJuniperMstEntry 31	Indicates the number of times memory allocation failures have occurred. This event generates a trap.
jnxMIMstRegionConfigChangeCount	jnxMIDot1sJuniperMstEntry 32	Indicates the number of times a Region Configuration Identifier Change was detected. This event generates a trap.
jnxMIMstCistBridgeRoleSelectionSemState	jnxMIDot1sJuniperMstEntry 33	Indicates the current state of the Port Role Selection State Machine of the bridge in a common spanning-tree context

Table 236: jnxMIDot1sJuniperMstTable (*continued*)

Object	Object Identifier	Description
jnxMIMstCistTimeSinceTopologyChange	jnxMIDot1sJuniperMstEntry 34	Indicates the time since the TcWhile Timer for any port of the Bridge was nonzero for the common spanning-tree context.
jnxMIMstCistTopChanges	jnxMIDot1sJuniperMstEntry 35	Indicates the number of times when there was at least one nonzero TcWhile Timer on the bridge for the common spanning-tree context.
jnxMIMstCistNewRootBridgeCount	jnxMIDot1sJuniperMstEntry 36	Indicates how many times the bridge has detected a root bridge change for a common-spanning-tree context. This event generates a trap.
jnxMIMstCistHelloTime	jnxMIDot1sJuniperMstEntry 37	Specifies the interval between the transmission of configuration BPDUs by this node on any port when it is either the root of the spanning-tree or trying to become the root.
jnxMIMstCistBridgeHelloTime	jnxMIDot1sJuniperMstEntry 38	Specifies the interval between the transmission of configuration bridge PDUs by this node.
jnxMIMstCistDynamicPathcostCalculation	jnxMIDot1sJuniperMstEntry 39	Indicates whether the dynamic path cost calculation is allowed. If set to true, path cost is calculated dynamically from the port speed; otherwise, the link speed at the time of port creation is used for calculating the path cost. In both cases, the user has configured a path cost for the port that is used. By default, dynamic path cost calculation is set to false.

jnxMIMstMstiBridgeTable

The **jnxMIMstMstiBridgeTable** provides information about a bridge that belongs to a given spanning-tree instance (MSTI). Each **jnxMIMstMstiBridgeEntry** points to a bridge in the spanning-tree instance and has the objects listed in [Table 237 on page 612](#).

Table 237: jnxMIMstMstiBridgeTable

Object	Object Identifier	Description
jnxMIMstMstiInstanceIndex	jnxMIMstMstiBridgeEntry 1	Identifies the spanning-tree instance to which the information belongs.

Table 237: jnxMIMstMstiBridgeTable (*continued*)

Object	Object Identifier	Description
jnxMIMstMstiBridgeRegionalRoot	jnxMIMstMstiBridgeEntry 2	Indicates the MSTI Regional Root Identifier value for the Instance. This value is used as the Regional Root Identifier parameter in all the configuration bridge PDUs originated by this node.
jnxMIMstMstiBridgePriority	jnxMIMstMstiBridgeEntry 3	Indicates the writable portion of the MSTI Bridge Identifier that comprises the first two octets. The values that are set for Bridge Priority must be in multiples of 4096.
jnxMIMstMstiRootCost	jnxMIMstMstiBridgeEntry 4	Indicates the cost of the path to the MSTI Regional Root as calculated by the bridge.
jnxMIMstMstiRootPort	jnxMIMstMstiBridgeEntry 5	Indicates the port number of the port that offers the lowest path cost from the bridge to the MSTI Region Root Bridge.
jnxMIMstMstiTimeSinceTopologyChange	jnxMIMstMstiBridgeEntry 6	Indicates the time (in hundredths of a second) since the TcWhile Timer for any port on this bridge was nonzero for this spanning-tree instance.
jnxMIMstMstiTopChanges	jnxMIMstMstiBridgeEntry 7	Indicates the number of times when there was at least one nonzero TcWhile Timer on the bridge for the spanning-tree instance.
jnxMIMstMstiNewRootBridgeCount	jnxMIMstMstiBridgeEntry 8	Indicates the number of times the bridge has detected a root bridge change for the spanning-tree instance. This event generates a trap.
jnxMIMstMstiBridgeRoleSelectionSemState	jnxMIMstMstiBridgeEntry 9	Shows the current state of the Port Role Selection State Machine for the spanning-tree instance of this bridge.
jnxMIMstInstanceUpCount	jnxMIMstMstiBridgeEntry 10	Indicates the number of times a new spanning-tree instance has been created. This counter is incremented whenever a new spanning-tree instance is created and also whenever a VLAN is mapped to the instance. This event generates a trap.
jnxMIMstInstanceDownCount	jnxMIMstMstiBridgeEntry 11	Indicates the number of times a spanning-tree instance has been deleted. This counter is incremented whenever a spanning-tree instance is deleted and also whenever a VLAN is unmapped from the instance. This event generates a trap.

Table 237: jnxMIMstMstiBridgeTable (*continued*)

Object	Object Identifier	Description
jnxMIMstOldDesignatedRoot	jnxMIMstMstiBridgeEntry 12	Indicates the bridge identifier of the old root of the spanning-tree instance as determined by the Spanning Tree Protocol.

jnxMIMstVlanInstanceMappingTable

The **jnxMIMstVlanInstanceMappingTable** contains information about the mapping between each instance of MSTP and associated VLANs. Each **jnxMIMstVlanInstanceMappingEntry** indicates the status and properties of a specific MSTP instance-VLAN mapping and has the objects listed in [Table 238 on page 614](#).

Table 238: jnxMIMstVlanInstanceMappingTable

Object	Object Identifier	Description
jnxMIMstInstanceIndex	jnxMIMstVlanInstanceMappingEntry 1	Identifies a multiple spanning-tree instance using an arbitrary integer from 1 through the value of Max Instance Number .
jnxMIMstMapVlanIndex	jnxMIMstVlanInstanceMappingEntry 2	Indicates that the VLAN ID is mapped to the multiple spanning-tree instance specified.
jnxMIMstUnMapVlanIndex	jnxMIMstVlanInstanceMappingEntry 3	Indicates that the VLAN ID is unmapped from the spanning-tree instance to which it was mapped.
jnxMIMstInstanceVlanMapped	jnxMIMstVlanInstanceMappingEntry 6	Represents a string of octets that contain one bit per VLAN. The first octet corresponds to VLANs with VLAN index values 1 through 8; the second octet to VLANs 9 through 16, and so on. The most significant bit of each octet corresponds to the lowest VlanIndex value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.
jnxMIMstInstanceVlanMapped2k	jnxMIMstVlanInstanceMappingEntry 7	Represents a string of octets that contain one bit per VLAN for VLANs with VlanIndex values from 1024 through 2047. The first octet corresponds to VLANs with VlanIndex values 1024 through 1031; the second octet to VLANs 1032 through 1039, and so on. The most significant bit of each octet corresponds to the lowest VlanIndex value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.

Table 238: jnxMIMstVlanInstanceMappingTable (*continued*)

Object	Object Identifier	Description
jnxMIMstInstanceVlanMapped3k	jnxMIMstVlanInstanceMappingEntry 8	Represents a string of octets that contain one bit per VLAN for VLANS with VlanIndex values from 2048 through 3071. The first octet corresponds to VLANs with VlanIndex values 2048 through 2055; the second octet to VLANs 2056 through 2063, and so on. The most significant bit of each octet corresponds to the lowest VlanIndex value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.
jnxMIMstInstanceVlanMapped4k	jnxMIMstVlanInstanceMappingEntry 9	Represents a string of octets that contain one bit per VLAN for VLANS with VlanIndex values from 3072 through 4095. The first octet corresponds to VLANs with VlanIndex values 3072 through 3079; the second octet to VLANs 3080 through 3087, and so on. The most significant bit of each octet corresponds to the lowest VlanIndex value in that octet. For each VLAN that is mapped to this MSTP instance, the bit corresponding to that VLAN is set to 1.

jnxMIMstCistPortTable

The **jnxMIMstCistPortTable** contains the information maintained by the ports of the Common and Internal Spanning Tree Protocol. [Table 239 on page 615](#) lists the parameters maintained by each **jnxMIMstCistPortEntry**.

Table 239: jnxMIMstCistPortTable

Object	Object Identifier	Description
jnxMIMstCistPort	jnxMIMstCistPortEntry 1	Specifies the port number of the port to which this entry is mapped.
jnxMIMstCistPortPathCost	jnxMIMstCistPortEntry 2	Indicates the contribution of this port to the path cost of paths towards the CIST root that includes this port.
jnxMIMstCistPortPriority	jnxMIMstCistPortEntry 3	Contains the four most significant bits of the Port Identifier of the spanning-tree instance that can be modified by setting the CistPortPriority value. The values that are set for Port Priority must be in multiples of 16.
jnxMIMstCistPortDesignatedRoot	jnxMIMstCistPortEntry 4	Specifies the unique Bridge Identifier that is recorded as the CIST root in the configuration bridge protocol data units (BPDUs).

Table 239: jnxMIMstCistPortTable (*continued*)

Object	Object Identifier	Description
jnxMIMstCistPortDesignatedBridge	jnxMIMstCistPortEntry 5	Specifies the unique Bridge Identifier of the bridge that is considered as the designated bridge for the port's segment.
jnxMIMstCistPortDesignatedPort	jnxMIMstCistPortEntry 6	Indicates the port identifier of the port on the designated bridge for this port's segment.
jnxMIMstCistPortAdminP2P	jnxMIMstCistPortEntry 7	Indicates the administrative point-to-point status of the LAN segment attached to this port. <ul style="list-style-type: none"> A value of forceTrue(0) indicates that this port must be treated as if it were connected to a point-to-point link. A value of forceFalse(1) indicates that this port should be treated as having a shared media connection. A value of auto(2) indicates that this port is considered to have a point-to-point link if it is an aggregator and all of its members are aggregatable, or if the MAC entity is configured for full duplex operation, either through autonegotiation or by management.
jnxMIMstCistPortOperP2P	jnxMIMstCistPortEntry 8	Indicates the operational point-to-point status of the LAN segment that is attached to this port. It indicates whether a port is considered to have a point-to-point connection or not. The value is determined by management or by autodetection, as described in the jnxMIMstCistPortAdminP2P object.
jnxMIMstCistPortAdminEdgeStatus	jnxMIMstCistPortEntry 9	Specifies the administrative value of the EdgePort parameter. A value of TRUE(1) indicates that this port must be treated as an edge port, and a value of FALSE(2) indicates that this port should be treated as a nonedge port.
jnxMIMstCistPortOperEdgeStatus	jnxMIMstCistPortEntry 10	Specifies the operational value of the EdgePort parameter. The object is initialized to the value of jnxMIMstCistPortAdminEdgeStatus and is set FALSE on reception of a BPDU.
jnxMIMstCistPortState	jnxMIMstCistPortEntry 12	Shows the current state of the port as defined by the Common Spanning Tree Protocol.
jnxMIMstCistPortForwardTransitions	jnxMIMstCistPortEntry 14	Indicates the number of times this port has transitioned to the forwarding state.
jnxMIMstCistPortRxMstBpduCount	jnxMIMstCistPortEntry 15	Indicates the number of multiple spanning-tree BPDUs that are received on this port.
jnxMIMstCistPortRxRstBpduCount	jnxMIMstCistPortEntry 16	Indicates the number of rapid spanning-tree BPDUs that are received on this port.

Table 239: jnxMIMstCistPortTable (*continued*)

Object	Object Identifier	Description
jnxMIMstCistPortRxConfigBpduCount	jnxMIMstCistPortEntry 17	Indicates the number of configuration BPDUs that are received on the port.
jnxMIMstCistPortRxTcnBpduCount	jnxMIMstCistPortEntry 18	Indicates the number of topology change notification BPDUs that are received on the port.
jnxMIMstCistPortTxMstBpduCount	jnxMIMstCistPortEntry 19	Indicates the number of multiple spanning-tree BPDUs that are transmitted from the port.
jnxMIMstCistPortTxRstBpduCount	jnxMIMstCistPortEntry 20	Indicates the number of rapid spanning-tree BPDUs that are transmitted from the port.
jnxMIMstCistPortTxConfigBpduCount	jnxMIMstCistPortEntry 21	Indicates the number of configuration BPDUs that are transmitted from the port.
jnxMIMstCistPortTxTcnBpduCount	jnxMIMstCistPortEntry 22	Indicates the number of TCN BPDUs that are transmitted from the port.
jnxMIMstCistPortInvalidMstBpduRxCount	jnxMIMstCistPortEntry 23	Indicates the number of invalid MST BPDUs that are received on the port.
jnxMIMstCistPortInvalidRstBpduRxCount	jnxMIMstCistPortEntry 24	Indicates the number of invalid RST BPDUs that are received on the port.
jnxMIMstCistPortInvalidConfigBpduRxCount	jnxMIMstCistPortEntry 25	Indicates the number of invalid configuration BPDUs that are received on the port.
jnxMIMstCistPortInvalidTcnBpduRxCount	jnxMIMstCistPortEntry 26	Indicates the number of invalid TCN BPDUs that are received on the port.
jnxMIMstCistPortTransmitSemState	jnxMIMstCistPortEntry 27	Indicates the current state of the port transmit state machine.
jnxMIMstCistPortReceiveSemState	jnxMIMstCistPortEntry 28	Indicates the current state of the port receive state machine.
jnxMIMstCistPortProtMigrationSemState	jnxMIMstCistPortEntry 29	Indicates the current state of the port protocol migration state machine.
jnxMIMstCistProtocolMigrationCount	jnxMIMstCistPortEntry 30	Indicates the number of times the port has migrated from one Spanning Tree Protocol version to another. The relevant protocols are STP-COMPATIBLE and RSTP/MSTP. This event generates a trap.
jnxMIMstCistPortDesignatedCost	jnxMIMstCistPortEntry 31	Indicates the path cost of the designated port of the segment connected to this port.
jnxMIMstCistPortRegionalRoot	jnxMIMstCistPortEntry 32	Specifies the unique Bridge Identifier of the bridge recorded as the CIST Regional Root Identifier in the configuration BPDUs transmitted.

Table 239: jnxMIMstCistPortTable (*continued*)

Object	Object Identifier	Description
jnxMIMstCistPortRegionalPathCost	jnxMIMstCistPortEntry 33	Specifies the contribution of this port to the path cost of paths towards the CIST regional root that includes this port.
jnxMIMstCistSelectedPortRole	jnxMIMstCistPortEntry 34	Indicates the selected port role of the port for this spanning-tree instance.
jnxMIMstCistCurrentPortRole	jnxMIMstCistPortEntry 35	Specifies the current port role of the port for this spanning-tree instance.
jnxMIMstCistPortInfoSemState	jnxMIMstCistPortEntry 36	Indicates the current state of the port information state machine for this port in this spanning-tree context.
jnxMIMstCistPortRoleTransitionSemState	jnxMIMstCistPortEntry 37	Indicates the current state of the port role transition state machine for this port in this spanning-tree context.
jnxMIMstCistPortStateTransitionSemState	jnxMIMstCistPortEntry 38	Indicates the current state of the port state transition state machine for this port in this spanning-tree context.
jnxMIMstCistPortTopologyChangeSemState	jnxMIMstCistPortEntry 39	Indicates the current state of the topology change state machine for this port in this spanning-tree context.
jnxMIMstCistPortHelloTime	jnxMIMstCistPortEntry 40	Indicates the interval between the transmission of configuration bridge PDUs on this port.
jnxMIMstCistPortOperVersion	jnxMIMstCistPortEntry 41	Indicates one of the following operational modes of the port: MSTP, RSTP, and STP-compatible.
jnxMIMstCistPortEffectivePortState	jnxMIMstCistPortEntry 42	Indicates the effective operational state of the port for CIST. This value is set to TRUE only when the port is operationally up in the Interface level and Protocol level for CIST. This value is set to FALSE for all other instances.
jnxMIMstCistPortAutoEdgeStatus	jnxMIMstCistPortEntry 43	Indicates one of the following states: <ul style="list-style-type: none"> • TRUE(1) when the detection of a port as edge port happens automatically • FALSE(2) when this feature is disabled.

jnxMIMstMstiPortTable

The **jnxMIMstMstiPortTable** contains information maintained by the nonCIST ports for each spanning-tree instance. Each **jnxMIMstMstiPortEntry** contains the parameters listed in [Table 240 on page 619](#).

Table 240: jnxMIMstMstiPortTable

Object	Object Identifier	Description
jnxMIMstMstiPort	jnxMIMstMstiPortEntry 1	Specifies the port number of the port to which this entry maps.
jnxMIMstMstiPortPathCost	jnxMIMstMstiPortEntry 2	Shows the contribution of this port to the path cost of paths toward the MSTI root that includes this port.
jnxMIMstMstiPortPriority	jnxMIMstMstiPortEntry 3	Enables you to specify the four most significant bits of the Port Identifier for a given spanning-tree instance, which can be modified independently for each spanning-tree instance supported by the bridge. The values that are set for Port Priority must be in multiples of 16.
jnxMIMstMstiPortDesignatedRoot	jnxMIMstMstiPortEntry 4	Indicates the unique Bridge Identifier of the bridge recorded as the MSTI regional root in the configuration BPDUs that are transmitted.
jnxMIMstMstiPortDesignatedBridge	jnxMIMstMstiPortEntry 5	Indicates the unique Bridge Identifier of the bridge that this port considers to be the designated bridge for the port's segment.
jnxMIMstMstiPortDesignatedPort	jnxMIMstMstiPortEntry 6	Indicates the port identifier of the port on the designated bridge for this port's segment.
jnxMIMstMstiPortState	jnxMIMstMstiPortEntry 7	Indicates the current state of the port as defined by the Multiple Spanning Tree protocol. A port that is in forwarding state in one instance can be in discarding (blocking) state in another instance.
jnxMIMstMstiPortForwardTransitions	jnxMIMstMstiPortEntry 9	Indicates the number of times this port has transitioned to the forwarding state for a specific instance.
jnxMIMstMstiPortReceivedBPDUs	jnxMIMstMstiPortEntry 10	Indicates the number of BPDUs received by this port for this spanning-tree instance.
jnxMIMstMstiPortTransmittedBPDUs	jnxMIMstMstiPortEntry 11	Indicates the number of BPDUs transmitted on this port for this spanning-tree instance.
jnxMIMstMstiPortInvalidBPDUsRcvd	jnxMIMstMstiPortEntry 12	Indicates the number of invalid BPDUs received on this port for this spanning-tree instance.
jnxMIMstMstiPortDesignatedCost	jnxMIMstMstiPortEntry 13	Indicates the path cost of the designated port of the segment connected to this port.
jnxMIMstMstiSelectedPortRole	jnxMIMstMstiPortEntry 14	Indicates the selected Port Role of the port for this spanning-tree instance.
jnxMIMstMstiCurrentPortRole	jnxMIMstMstiPortEntry 15	Indicates the current Port Role of the port for this spanning-tree instance.

Table 240: jnxMIMstMstiPortTable (continued)

Object	Object Identifier	Description
jnxMIMstMstiPortInfoSemState	jnxMIMstMstiPortEntry 16	Shows the current state of the port information state machine for this port in this spanning-tree context.
jnxMIMstMstiPortRoleTransitionSemState	jnxMIMstMstiPortEntry 17	Shows the current state of the port role transition state machine for this port in this spanning-tree context.
jnxMIMstMstiPortStateTransitionSemState	jnxMIMstMstiPortEntry 18	Shows the current state of the port state transition state machine for this port in this spanning-tree context.
jnxMIMstMstiPortTopologyChangeSemState	jnxMIMstMstiPortEntry 19	Shows the current state of the topology change state machine for this port in this spanning-tree context.
jnxMIMstMstiPortEffectivePortState	jnxMIMstMstiPortEntry 20	Shows the effective operational state of the port for the specific instance. The value is set to TRUE only when the port is operationally up in the interface level and protocol level for the specific instance. This is set to FALSE at all other times.

Juniper Networks Enterprise-Specific MIMSTP Traps

Table 241 on page 620 lists the enterprise-specific MIMSTP traps.

Table 241: Enterprise-Specific MIMSTP Traps

Object	Object Identifier	Description
jnxMIMstGenTrap	jnxMIMstTraps 1	Generated when any of the general events such as protocol up or protocol down occurs.
jnxMIMstErrTrap	jnxMIMstTraps 2	Generated when any of the error events such as a memory failure, buffer failure, protocol migration, or new root or topology change occurs.
jnxMIMstNewRootTrap	jnxMIMstTraps 3	Generated when a new root bridge is selected in the topology. The jnxMIMstNewRootTrap indicates that the sending agent has become the new root of the spanning-tree. The trap is sent by a bridge soon after its election as the new root.
jnxMIMstTopologyChgTrap	jnxMIMstTraps 4	Generated when a topology change is detected.
jnxMIMstProtocolMigrationTrap	jnxMIMstTraps 5	Generated when a port protocol migration happens on the port.
jnxMIMstInvalidBpduRxdTrap	jnxMIMstTraps 6	Generated when an invalid packet is received for bpdu/stp/rstp/maximum age/forward delay/hello time .
jnxMIMstRegionConfigChangeTrap	jnxMIMstTraps 7	Generated when the multiple spanning-tree region's configuration identifier changes.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 47

MPLS LDP MIB

- Interpreting the Enterprise-Specific MPLS LDP MIB on page 623

Interpreting the Enterprise-Specific MPLS LDP MIB

- MPLS LDP MIB on page 623

MPLS LDP MIB

The Juniper Networks enterprise-specific MPLS LDF MIB, whose object identifier is **{jnxMibs 36}**, contains object definitions as described in RFC 3815, *Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)*.

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

For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls-ldp.txt.

Related Documentation

- Standard SNMP MIBs Supported by Junos OS on page 9
- Juniper Networks Enterprise-Specific MIBs on page 25
- Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36
- SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46
- Standard SNMP Traps Supported on Devices Running Junos OS on page 57
- Juniper Networks Enterprise-Specific SNMP Traps on page 57

CHAPTER 48

MPLS MIB

- Interpreting the Enterprise-Specific MPLS MIB on page 625

Interpreting the Enterprise-Specific MPLS MIB

- MPLS MIB on page 625
- mplsInfo on page 625
- mplsTEInfo on page 626
- mplsAdminGroup on page 626
- mplsLsplInfoList on page 627
- Juniper Networks Enterprise-Specific MPLS Traps on page 630

MPLS MIB

The Juniper Networks enterprise-specific MPLS MIB, whose object identifier is **{jnxMibs 2}**, provides information about MPLS paths and defines MPLS notifications.

The table **mplsLspList** and the sequence for **mplsLspEntry** have been deprecated and replaced by a new table **mplsLsplInfoList** to extend support for LSP names longer than 32 characters.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls.txt.

For information about the enterprise-specific MPLS MIB objects, see the following topics:

- mplsInfo on page 625
- mplsTEInfo on page 626
- mplsAdminGroup on page 626
- mplsLsplInfoList on page 627
- Juniper Networks Enterprise-Specific MPLS Traps on page 630

mplsInfo

The **mplsInfo** table, whose object identifier is **{mpls 1}**, contains the objects listed in Table 242 on page 626.

Table 242: mplsInfo

Object	Object ID	Description
mplsVersion	mplsInfo 1	Shows the MPLS version number.
mplsSignalingProto	mplsInfo 2	Indicates the MPLS signaling protocol. This object uses the following integer values to denote the MPLS signaling protocol: <ul style="list-style-type: none"> • 1—None • 2—Other • 3—RSVP • 4—LDP
mplsConfiguredLsps	mplsInfo 3	Indicates the number of LSPs configured on the router.
mplsActiveLsps	mplsInfo 4	Indicates the number of active LSPs on the router.

mplsTEInfo

The **mplsTEInfo** table, whose object identifier is **{mpls 2}**, contains the objects listed in [Table 243 on page 626](#).

Table 243: mplsTEInfo

Object	Object ID	Description
mplsTEDistProtocol	mplsTEInfo 1	Indicates the Interior Gateway Protocol (IGP) used to distribute Traffic Engineering information and topology to each Label Switch Router (LSR) for automatic path computation. This object uses the following integer values to denote the protocols: <ul style="list-style-type: none"> • 1—None • 2—IS-IS • 3—OSPF • 4—IS-IS – OSPF
mplsAdminGroupList	mplsTEInfo 2	Contains the list of administrative groups configured on the router. Administrative groups are used to label links in the Traffic Engineering topology for specifying constraints (include and exclude) on LSP paths.

mplsAdminGroup

The **mplsAdminGroup**, whose object identifier is **{mplsAdminGroupList 1}**, contains **mplsAdminGroupNumber** and **mplsAdminGroupName** objects, and provides a mapping between the group number and name.

- **mplsAdminGroupNumber**—Whose object identifier is **{mplsAdminGroup 1}**, contains an integer value from 0 through 31. This object indexes the **mplsAdminGroup**.
- **mplsAdminGroupName**—Whose object identifier is **{mplsAdminGroup 2}**, contains the name of the **mplsAdminGroup**. This object can contain a string of not more than 16 characters.

mplsLspInfoList

The **mplsLspInfoList**, whose object identifier is **{mpls 5}**, contains a list of Label Switched Paths (LSPs) configured on the router. Each **mplsLspInfoEntry** contains information about a particular LSP, and has the objects listed in [Table 244 on page 627](#).

Table 244: mplsLspInfoEntry

Object	Object ID	Description
mplsLspInfoName	mplsLspInfoEntry 1	Contains the name of the LSP. This object can contain a string of not more than 64 characters.
mplsLspInfoState	mplsLspInfoEntry 2	Contains one of the following integer values to indicate the operational state of the LSP: <ul style="list-style-type: none"> • 1—Unknown • 2—Up • 3—Down • 4—notInService: Indicates that the LSP has been torn down or never been signaled due to the lack of demand for its protection. • 5—backupActive: indicates that the LSP is up and carrying user traffic for at least one protected LSP due to the failure of the LSP, which has caused the creation of a backup LSP.
mplsLspInfoOctets	mplsLspInfoEntry 3	Indicates the number of octets that have been forwarded over the current LSP active path. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object might not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsLspInfoPackets	mplsLspInfoEntry 4	Indicates the number of packets that have been forwarded over the current LSP active path. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object might not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsLspInfoAge	mplsLspInfoEntry 5	Indicates the time duration (in 10-millisecond intervals) since the inception of the LSP.
mplsLspInfoTimeUp	mplsLspInfoEntry 6	Indicates the total time (in 10-millisecond intervals) that the LSP has been operational. The percentage of up time can be calculated using the following formula: mplsLspInfoTimeUp/mplsLspInfoAge x 100.
mplsLspInfoPrimaryTimeUp	mplsLspInfoEntry 7	Indicates the total time (in 10-millisecond intervals) that the primary path of the LSP has been operational.
mplsLspInfoTransitions	mplsLspInfoEntry 8	Indicates the number of state transitions, from up to down and down to up, that the LSP has undergone.

Table 244: mplsLspInfoEntry (continued)

Object	Object ID	Description
mplsLspInfoLastTransition	mplsLspInfoEntry 9	Shows the time (in 10-millisecond intervals) since the last state transition occurred on the LSP.
mplsLspInfoPathChanges	mplsLspInfoEntry 10	Shows the number of path changes that occurred on the LSP. Every path change (path down, path up, and path change) generates a syslog entry or trap or both if the corresponding configuration is enabled.
mplsLspInfoLastPathChange	mplsLspInfoEntry 11	Indicates the time (in 10-millisecond intervals) since the last path change occurred on the LSP.
mplsLspInfoConfiguredPaths	mplsLspInfoEntry 12	Indicates the number of paths configured for the LSP.
mplsLspInfoStandbyPaths	mplsLspInfoEntry 13	Indicates the number of standby paths configured on the LSP.
mplsLspInfoOperationalPaths	mplsLspInfoEntry 14	Indicates the number of operational paths for the LSP. The value of this object includes the currently active path as well as the operational standby paths.
mplsLspInfoFrom	mplsLspInfoEntry 15	Contains the source IP address of the LSP.
mplsLspInfoTo	mplsLspInfoEntry 16	Contains the destination IP address of the LSP.
mplsPathInfoName	mplsLspInfoEntry 17	Shows the name of the active path for the LSP. If the path does not have a name, the mplsLspInfoEntry objects listed in this table are invalid.
mplsPathInfoType	mplsLspInfoEntry 18	<p>Contains one of the following integer values to denote the type of the active path:</p> <ul style="list-style-type: none"> • 1—Other • 2—Primary • 3—Standby • 4—Secondary • 5—Bypass <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>

Table 244: mplsLspInfoEntry (continued)

Object	Object ID	Description
mplsPathInfoExplicitRoute	mplsLspInfoEntry 19	<p>Contains the explicit route used to set up the LSP. The explicit router can be one configured by the user or a generated route that satisfies the constraints set by the user.</p> <p>The value of this object is stored in the following format: xxx.xxx.xxx.xxx S/L, where S/L stands for Strict/Loose route. Each explicit route appears in a new line.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoRecordRoute	mplsLspInfoEntry 20	<p>Shows the route actually used for the LSP as recorded by the signaling protocol.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoBandwidth	mplsLspInfoEntry 21	<p>Indicates the configured bandwidth (in kbps) for the LSP.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoCOS	mplsLspInfoEntry 22	<p>Indicates the class of service (CoS) configured for the path. If the value of this object is from 0 through 7, it goes in the 3-bit CoS field in the label. If the value is 255, the value in the CoS field of the label depends on other factors.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoInclude	mplsLspInfoEntry 23	<p>Contains a configured set of colors represented by bit vector. For each link this path goes through, the link must have colors associated with the path, and the intersection of the link's colors and the include set must be set to a value other than null.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoExclude	mplsLspInfoEntry 24	<p>Contains a configured set of colors represented by bit vector. For each link the path goes through, the link must have colors associated with the path, and the intersection of the link's colors and the exclude set must be set to null.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>
mplsPathInfoSetupPriority	mplsLspInfoEntry 25	<p>Indicates the setup priority configured for the path. This object contains integer values from 0 through 7.</p> <p>NOTE: The value of this object is invalid if mplsPathInfoName is blank.</p>

Table 244: mplsLspInfoEntry (*continued*)

Object	Object ID	Description
mplsPathInfoHoldPriority	mplsLspInfoEntry 26	Indicates the hold priority configured for the path. This object contains integer values from 0 through 7. NOTE: The value of this object is invalid if mplsPathInfoName is blank.
mplsPathInfoProperties	mplsLspInfoEntry 27	Denotes the properties configured for the path. This value is represented as a bit map. The possible values are: <ul style="list-style-type: none"> 1—Record-Route 2—Adaptive 4—CSPF 8—Mergeable 16—Preemptable 32—Preemptive 64—Fast-Reroute NOTE: The value of this object is invalid if mplsPathInfoName is blank.
mplsLspInfoAggrOctets	mplsLspInfoEntry 28	Indicates the total number of octets that have been forwarded over all instances of the current LSP. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object might not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsLspInfoAggrPackets	mplsLspInfoEntry 29	Indicates the total number of packets that have been forwarded over all instances of the current LSP. Because the MPLS statistics are collected only at predefined intervals (default of 5 minutes), the value of this object might not reflect the real-time value. The value of the object is not updated if MPLS statistics collection is not enabled.
mplsPathInfoRecordRouteWithLabels	mplsLspInfoEntry 30	Indicates the route taken by the path along with the label information.

Juniper Networks Enterprise-Specific MPLS Traps

Table 245 on page 630 lists the enterprise-specific MPLS traps based on **mplsLspInfoName**.

Table 245: MPLS Traps

Object	Object ID	Description
mplsLspInfoUp	mplsLspTraps 1	Indicates that the LSP (mplsLspInfoName) is up. The current active path is represented by mplsPathInfoName .
mplsLspInfoDown	mplsLspTraps 2	Indicates that the LSP (mplsLspInfoName) is down because the current active path (mplsPathInfoName) has gone down.

Table 245: MPLS Traps (*continued*)

Object	Object ID	Description
<code>mplsLspInfoChange</code>	<code>mplsLspTraps 3</code>	Indicates that the LSP (<code>mplsLspInfoName</code>) has switched traffic to a new active path (<code>mplsPathInfoName</code>) without changing the state (up) before or after the switch.
<code>mplsLspInfoPathDown</code>	<code>mplsLspTraps 4</code>	Indicates that the specified path (<code>mplsPathInfoName</code>) for the LSP (<code>mplsLspInfoName</code>) has gone down.
<code>mplsLspInfoPathUp</code>	<code>mplsLspTraps 5</code>	Indicates that the specified path (<code>mplsPathInfoName</code>) for the LSP (<code>mplsLspInfoName</code>) has come up.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

NAT Objects MIB

- Interpreting the Enterprise-Specific Network Address Translation Objects MIB on page 633

Interpreting the Enterprise-Specific Network Address Translation Objects MIB

- NAT Objects MIB on page 633
- Source NAT Table on page 634
- New Source NAT Table on page 635
- jnxJsNatIfSrcPoolPortTable on page 636
- NAT Rule Hit Table on page 637
- NAT Pool Hit Table on page 637
- Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways on page 638

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/junos13.2/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 634
- New Source NAT Table on page 635
- jnxJsNatIfSrcPoolPortTable on page 636

- [NAT Rule Hit Table on page 637](#)
- [NAT Pool Hit Table on page 637](#)
- [Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways on page 638](#)

Source NAT Table

Table 246 on page 634 identifies objects in the Source NAT Table.

Table 246: Source NAT Table

Object	Object ID	Description
<code>jnxJsSrcNatNumOfEntries</code>	<code>jnxJsNatObjects 1</code>	Total number of dynamic addresses being translated.
<code>jnxJsSrcNatTable</code>	<code>jnxJsNatObjects 2</code>	<p>Table that exposes the source NAT attributes of the translated addresses.</p> <p>When performing source IP address translation, the device translates the original source IP address or port number to a different one. The resource, address source pools, provides the security device with a supply of addresses from which to draw when performing source NAT.</p> <p>The security device has the following types of source pools:</p> <ul style="list-style-type: none"> • Source pool with Port Address Translation (PAT) • Source pool without PAT • Static source pool <p>This table contains information about source IP address translation only.</p>
<code>jnxJsSrcNatEntry</code>	<code>jnxJsSrcNatTable 1</code>	<p>Source NAT address entries. This object is indexed by the address pool table and the address allocated.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • <code>jnxJsNatSrcIpPoolName</code> • <code>jnxJsNatSrcGlobalAddr</code> • <code>jnxJsNatSrcPortPoolType</code> • <code>jnxJsNatSrcNumOfPortInuse</code> • <code>jnxJsNatSrcNumOfSessions</code> • <code>jnxJsNatSrcAssociatedIf</code>
<code>jnxJsNatSrcIpPoolName</code>	<code>jnxJsSrcNatEntry 1</code>	Address pool from which the translated address is allocated.
<code>jnxJsNatSrcGlobalAddr</code>	<code>jnxJsSrcNatEntry 2</code>	Name of dynamic source IP address allocated from the address pool used in the NAT translation.

Table 246: Source NAT Table (*continued*)

Object	Object ID	Description
<code>jnxJsNatSrcPortPoolType</code>	<code>jnxJsSrcNatEntry 3</code>	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: <ul style="list-style-type: none"> • 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.
<code>jnxJsNatSrcNumOfPortInuse</code>	<code>jnxJsSrcNatEntry 4</code>	Number of ports in use for this NAT address entry. This attribute is applicable to only NAT translation with PAT.
<code>jnxJsNatSrcNumOfSessions</code>	<code>jnxJsSrcNatEntry 5</code>	Number of sessions in use for this NAT address entry. This attribute is applicable to only NAT translation without PAT.
<code>jnxJsNatSrcAssociatedIf</code>	<code>jnxJsSrcNatEntry 6</code>	Index of interfaces associated with this NAT address entry. For each interface, the value is a unique value, greater than zero.

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 247 on page 635](#) identifies objects in the New Source NAT Table.

Table 247: New Source NAT Table

Object	Object ID	Description
<code>jnxJsSrcNatStatsEntry</code>	<code>jnxJsSrcNatStatsTable 1</code>	Source NAT address entries. This object is indexed by the address pool table and the allocated address. Sequence of parameters: <ul style="list-style-type: none"> • <code>jnxJsNatSrcPoolName</code> • <code>jnxJsNatSrcXlatedAddrType</code> • <code>jnxJsNatSrcXlatedAddr</code> • <code>jnxJsNatSrcPoolType</code> • <code>jnxJsNatSrcNumPortInuse</code> • <code>jnxJsNatSrcNumSessions</code>
<code>jnxJsNatSrcPoolName</code>	<code>jnxJsSrcNatStatsEntry 1</code>	The name of the dynamic source IP address pool. This is the address pool from which the translated address is allocated.

Table 247: New Source NAT Table (*continued*)

Object	Object ID	Description
<code>jnxJsNatSrcXlatedAddrType</code>	<code>jnxJsSrcNatStatsEntry 2</code>	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.
<code>jnxJsNatSrcXlatedAddr</code>	<code>jnxJsSrcNatStatsEntry 3</code>	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.
<code>jnxJsNatSrcPoolType</code>	<code>jnxJsSrcNatStatsEntry 4</code>	<p>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:</p> <ul style="list-style-type: none"> • 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.
<code>jnxJsNatSrcNumPortInuse</code>	<code>jnxJsSrcNatStatsEntry 5</code>	The number of ports in use for this NAT address entry. This attribute is applicable only to NAT translation with PAT.
<code>jnxJsNatSrcNumSessions</code>	<code>jnxJsSrcNatStatsEntry 6</code>	The number of sessions in use based on this NAT address entry. This attribute is applicable only to NAT translation without PAT.

`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 248 on page 637](#).

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 248: jnxJsNatIfSrcPoolPortTable

Object	Object ID	Description
jnxJsNatIfSrcPoolPortEntry	jnxJsNatIfSrcPoolPortTable 1	Source NAT address entries. This object is indexed by the address pool table and the address. Sequence of parameters: <ul style="list-style-type: none"> • 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.

NAT Rule Hit Table

The **jnxJsNatRuleTable**, whose object ID is {**jnxJsNatObjects 5**}, monitors the NAT rule hits. [Table 249 on page 637](#) identifies objects in the NAT Rule Hit Table.

Table 249: 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: <ul style="list-style-type: none"> • 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.

NAT Pool Hit Table

The **jnxJsNatPoolTable**, whose object ID is {**jnxJsNatObjects 6**}, monitors the NAT pool hits. [Table 250 on page 638](#) identifies objects in the NAT Pool Hit Table.

Table 250: NAT Pool Hit Table

Object	Object ID	Description
<code>jnxJsNatPoolEntry</code>	<code>jnxJsNatPoolTable 1</code>	NAT pool hit entries. This object is indexed by the pool index. Sequence of parameters: <ul style="list-style-type: none"> • <code>jnxJsNatPoolName</code> • <code>jnxJsNatPoolType</code> • <code>jnxJsNatPoolTransHits</code>
<code>jnxJsNatPoolName</code>	<code>jnxJsNatPoolEntry 1</code>	NAT pool name.
<code>jnxJsNatPoolType</code>	<code>jnxJsNatPoolEntry 2</code>	NAT types such as source, destination, and static.
<code>jnxJsNatPoolTransHits</code>	<code>jnxJsNatPoolEntry 3</code>	Number of hits on this NAT pool.

Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways

Table 251 on page 638 lists NAT trap definition objects on SRX Series Services Gateways.

Table 251: NAT Trap Definitions

Object	Object ID	Description
<code>jnxJsNatAddrPoolThresholdStatus</code>	<code>jnxJsNatNotifications 1</code>	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. <code>jnxJsNatSrcIpPoolName</code> is the name of the resource pool. <code>jnxJsNatAddrPoolUtil</code> is the percentage of utilization of the address pool.
<code>jnxJsNatAddrPoolUtil</code>	<code>jnxJsNatTrapVars 1</code>	Dynamic address pool utilization expressed as a percentage.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 50

Network Address Translation Resources—Monitoring MIB

- Interpreting the Enterprise-Specific Network Address Translation Resources—Monitoring MIB on page 639

Interpreting the Enterprise-Specific Network Address Translation Resources—Monitoring MIB

- Network Address Translation Resources—Monitoring MIB on page 639
- Source NAT Table on page 640
- NAT Pool Hit Table on page 641
- NAT Rule Hit Table on page 641
- Juniper Networks Enterprise-Specific NAT Traps on MX Series 3D Universal Edge Routers on page 642

Network Address Translation Resources—Monitoring MIB

The Juniper Networks enterprise-specific SNMP MIB **jnxNatMIB** defines the objects that are used to monitor Network Address Translation (NAT) pools usage and NAT rules. The object ID of the MIB is **{jnxNAT 1}** and it is under the **JNX-SVCS-SMI** hierarchy.

The following MIB objects are supported by **JUNIPER-NAT-MIB** for implementing SNMP MIBs to monitor NAT resources on the Multiservices PIC and Multiservices DPC:

- **jnxNatNotifications** {jnxNatMIB 0}
- **jnxNatObjects** {jnxNatMIB 1}
- **jnxNatTrapVars** {jnxNatMIB 2}

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sp-nat.txt.

For information about enterprise-specific SNMP NAT MIBs for monitoring NAT resources, see the following topics:

- [NAT Pool Hit Table on page 641](#)
- [NAT Rule Hit Table on page 641](#)
- [Juniper Networks Enterprise-Specific NAT Traps on MX Series 3D Universal Edge Routers on page 642](#)
- [Source NAT Table on page 640](#)

Source NAT Table

The **jnxSrcNatStatsTable** table, whose object ID is **{jnxNatObjects 1}**, lists the source NAT attributes of the translated addresses.

When performing source IP address translation, the services PIC translates the original source IP address with or without the port number to a different address. The resource, that is, the address source pools provide the services PIC with a supply of addresses from which an address is selected to perform the source network address translation.

[Table 252 on page 640](#) identifies objects in the **jnxNatPoolPortTable** branch. It contains information about source IP address translation only.

Table 252: Source NAT Table

Object	Object ID	Description
jnxSrcNatStatsEntry	jnxSrcNatStatsTable 1	The source NAT address entries. The source NAT address entries are indexed by the address pool table and the address allocated.
jnxNatSrcPoolName	jnxSrcNatStatsEntry 1	The name of dynamic source IP address pool. The translated address is allocated from this address pool.
jnxNatSrcXlatedAddrType	jnxSrcNatStatsEntry 2	The type of dynamic source IP address allocated from the address pool, which is used in the NAT translation. IPv4 and IPv6 addresses are supported for NAT MIB.
jnxNatSrcPoolType	jnxSrcNatStatsEntry 3	The source port pool type indicates whether the address translation is done with the port or without the port, or if it is a static translation. The source NAT can do an address translation with or without the port translation.
jnxNatSrcNumPortAvail	jnxSrcNatStatsEntry 4	Indicates the number of ports available with this pool.
jnxNatSrcNumPortInuse	jnxSrcNatStatsEntry 5	Indicates the number of ports in use for a given NAT address entry. This attribute is applicable only to the source NAT translation with port number translation.

Table 252: Source NAT Table (*continued*)

Object	Object ID	Description
jnxNatSrcNumAddressAvail	jnxSrcNatStatsEntry 6	Indicates the total number of addresses available in this pool.
jnxNatSrcNumAddressInUse	jnxSrcNatStatsEntry 7	Indicates the number of addresses in use from this pool. This attribute is applicable only to the pools used with source dynamic translations.
jnxNatSrcNumSessions	jnxSrcNatStatsEntry 8	This NAT address entry indicates the number of sessions in use.

NAT Pool Hit Table

The **jnxNatPoolTable** table, whose object ID is **{jnxNatObjects 3}**, monitors NAT pool hits.

[Table 253 on page 641](#) identifies objects in the **jnxNatPoolPortTable** branch.

Table 253: NAT Pool Hit Table

Object	Object ID	Description
jnxNatPoolEntry	jnxNatPoolTable 1	Specifies the NAT pool hit entries. The NAT pool hit entries are indexed by the pool index.
jnxNatPoolName	jnxNatPoolEntry 1	Specifies the NAT pool name.
jnxNatPoolType	jnxNatPoolEntry 2	Indicates the NAT types as one of the following: <ul style="list-style-type: none"> • Source • Destination • Static
jnxNatPoolTransHits	jnxNatPoolEntry 3	Indicates the number of hits in the specified NAT pool.

NAT Rule Hit Table

The **jnxNatRuleTable** table, whose object ID is **{jnxNatObjects 2}**, monitors the NAT rule hits.

[Table 254 on page 641](#) identifies objects in the **jnxNatRuleTable** branch.

Table 254: NAT Rule Hit Table

Object	Object ID	Description
jnxNatRuleEntry	jnxNatRuleTable 1	Specifies NAT rule hit entries. The NAT rule hit entries are indexed by the rule index.
jnxNatRuleName	jnxNatRuleEntry 1	Specifies the NAT rule name.

Table 254: NAT Rule Hit Table (*continued*)

Object	Object ID	Description
<code>jnxNatRuleType</code>	<code>jnxNatRuleEntry 2</code>	Indicates the NAT types as one of the following: <ul style="list-style-type: none"> • Source • Destination • Static
<code>jnxNatRuleTransHits</code>	<code>jnxNatRuleEntry 3</code>	Indicates the number of hits in this NAT rule.

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

The `jnxNatAddrPoolThresholdStatus` and `jnxNatAddrPoolUtil` objects notify the user about the usage of NAT resources in percentage.

Table 255 on page 642 lists the NAT trap definition objects.

Table 255: NAT Trap Definitions

Object	Object ID	Description
<code>jnxNatAddrPoolUtil</code>	<code>jnxNatTrapVars 1</code>	Displays the dynamic address pool utilization in percentage.
<code>jnxNatTrapSrcPoolName</code>	<code>jnxNatTrapVars 2</code>	Displays the source NAT pool name that issues the trap.
<code>jnxNatAddrPoolThresholdStatus</code>	<code>jnxNatNotifications 1</code>	<p>The NAT address pool utilization threshold status trap indicates that the address pool utilization has either exceeded a certain set percentage or is lower than a set percentage.</p> <p><code>jnxNatTrapPoolName</code> is the name of the resource pool.</p> <p><code>jnxNatAddrPoolUtil</code> is the percentage of utilization of the address pool.</p>

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

OTN Interface Management MIB

- [Interpreting the Enterprise-Specific OTN Interface Management MIB on page 643](#)

Interpreting the Enterprise-Specific OTN Interface Management MIB

- [OTN Interface Management MIB on page 643](#)
- [OTN Interface Management MIB Textual Conventions on page 644](#)
- [jnxOtnAlarmTable on page 645](#)
- [jnxOtnCurrentOdu15minTable on page 645](#)

OTN Interface Management MIB

The Juniper Networks enterprise-specific Optical Transport Network (OTN) Interface Management MIB, whose object identifier is **{jnxMibs 56}**, extends the standard OTN Interface Management MIB and defines the objects for managing OTN interfaces for Juniper Networks devices.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-otn.txt.

For information about the enterprise-specific OTN Interface Management MIB objects, see the following topics:

- [OTN Interface Management MIB Textual Conventions on page 644](#)
- [jnxOtnAlarmTable on page 645](#)
- [jnxOtnCurrentOdu15minTable on page 645](#)
- *Optical Parameters Associated with Black Link End Points*
- *Managed objects for the Optics Interface*
- *Managed Objects for the OTN Interface*

OTN Interface Management MIB Textual Conventions

JnxOtnAlarmId specifies the SONET/SDH alarms present on an interface. The possible values are:

- **otnLosAlarm (0)**—OTN loss of signal alarm
- **otnLofAlarm (1)**—OTN loss of frame alarm
- **otnLomAlarm (2)**—OTN loss of multiframe alarm
- **otnWavelengthlockAlarm (3)**—OTN wavelength lock alarm
- **otnOtuAisAlarm (4)**—OTN OTU AIS alarm
- **otnOtuBdiAlarm (5)**—OTN OTU BDI alarm
- **otnOtuTtimAlarm (6)**—OTN OTU TTIM alarm
- **otnOtuIaeAlarm (7)**—OTN OTU IAE alarm
- **otnOtuSdAlarm (8)**—OTN OTU bit error rate defect alarm
- **otnOtuFecExcessiveErrsAlarm (10)**—OTN OTU FEC excessive errors alarm
- **otnOtuFecDegradedErrsAlarm (11)**—OTN OTU FEC degraded errors alarm
- **otnOtuBbeThresholdAlarm (12)**—OTN OTU BBE threshold alarm
- **otnOtuEsThresholdAlarm (13)**—OTN OTU ES threshold alarm
- **otnOtuSesThresholdAlarm (14)**—OTN OTU SES threshold alarm
- **otnOtuUasThresholdAlarm (15)**—OTN OTU UAS threshold alarm
- **otnOduAisAlarm (16)**—OTN ODU AIS alarm
- **otnOduOciAlarm (17)**—OTN ODU OCI alarm
- **otnOduLckAlarm (18)**—OTN ODU LCK alarm
- **otnOduBdiAlarm (19)**—OTN ODU BDI alarm
- **otnOduTtimAlarm (20)**—OTN ODU TTIM alarm
- **otnOduSdAlarm (21)**—OTN ODU bit error rate defect alarm
- **otnOduSfAlarm (22)**—OTN ODU bit error rate fault alarm
- **otnOduRxUpsChange (23)**—OTN Rx APS change
- **otnOduBbeThresholdAlarm (24)**—OTN ODU BBE threshold alarm
- **otnOduEsThresholdAlarm (25)**—OTN OTU ES threshold alarm
- **otnOduSesThresholdAlarm (26)**—OTN OTU SES threshold alarm
- **otnOduUasThresholdAlarm (27)**—OTN ODU UAS threshold alarm
- **otnOpuPMTAlarm (28)**—OTN OPU payload mismatch alarm

jnxOtnAlarmTable

jnxOtnAlarmTable, whose object ID is **{jnxOtnAlarms1}**, contains information about alarms on the SONET/SDH physical interfaces on a device running the Junos OS. Alarm information is stored in **jnxOtnAlarmEntry**. Each **jnxOtnAlarmEntry**, whose object identifier is **{jnxOtnAlarmTable 1}**, contains the objects listed in [Table 256 on page 645](#).

Table 256: jnxOtnAlarmEntry

Object	Object ID	Description
jnxOtnCurrentAlarms	jnxOtnAlarmEntry 1	Identifies the active OTN alarms on the interface.
jnxOtnLastAlarmId	jnxOtnAlarmEntry 2	Identifies the latest OTN alarm that was set or cleared on the interface.
jnxOtnLastAlarmTime	jnxOtnAlarmEntry 3	Shows the sysUpTime value when the management system was informed of the last alarm event.
jnxOtnLastAlarmDate	jnxOtnAlarmEntry 4	Shows the date and time when the management system was informed about the last alarm event.
jnxOtnLastAlarmEvent	jnxOtnAlarmEntry 5	Indicates whether the last alarm event was to set a new alarm or to clear a previously set alarm.

jnxOtnCurrentOdu15minTable

jnxOtnCurrentOdu15minTable, whose object identifier is **{jnxOtnPerformanceMonitoring 1}**, stores information about the optical data unit (ODU) performance monitoring. Each **jnxOtnCurrentOdu15minEntry**, whose object identifier is **{jnxOtnCurrentOdu15minTable 1}**, contains the objects listed in [Table 257 on page 645](#).

Table 257: jnxOtnCurrentOdu15minEntry

Object	Object ID	Description
jnxOtnCurrentOdu15minBIP	jnxOtnCurrentOdu15minEntry 1	Shows the current 15-minute BIP counter.
jnxOtnCurrentOdu15minBBE	jnxOtnCurrentOdu15minEntry 2	Shows the current 15-minute BBE counter.
jnxOtnCurrentOdu15minES	jnxOtnCurrentOdu15minEntry 3	Shows the current 15-minute ES counter.
jnxOtnCurrentOdu15minSES	jnxOtnCurrentOdu15minEntry 4	Shows the current 15-minute SES counter.
jnxOtnCurrentOdu15minUAS	jnxOtnCurrentOdu15minEntry 5	Shows the current 15-minute UAS counter.
jnxOtnCurrentOdu15minElapsedTime	jnxOtnCurrentOdu15minEntry 6	Shows the time elapsed so far in the 15-minute interval.

Related Documentation • [Standard SNMP MIBs Supported by Junos OS on page 9](#)

- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Packet Forwarding Engine MIB

- Interpreting the Enterprise-Specific Packet Forwarding Engine MIB on page 647

Interpreting the Enterprise-Specific Packet Forwarding Engine MIB

- Packet Forwarding Engine MIB on page 647
- jnxPfeNotifyGTable on page 647
- jnxPfeNotifyTypeTable on page 649

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/junos13.2/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:

- jnxPfeNotifyGTable on page 647
- jnxPfeNotifyTypeTable on page 649

jnxPfeNotifyGTable

jnxPfeNotifyGTable contains global Packet Forwarding Engine notification statistics.

JnxPfeNotifyGEntry contains notification statistics for each Packet Forwarding Engine slot. Each entry has objects listed in [Table 258 on page 648](#).

Table 258: jnxPfeNotifyGIEEntry

Object	Object Identifier	Description
jnxPfeNotifyGISlot	jnxPfeNotifyGIEEntry 1	The slot number for a set of Packet Forwarding Engine notification statistics.
jnxPfeNotifyGIParsed	jnxPfeNotifyGIEEntry 2	The number of notifications reported by the Packet Forwarding Engine controller, which manages packet forwarding functions.
jnxPfeNotifyGIAged	jnxPfeNotifyGIEEntry 3	The number of notifications dropped because they have expired.
jnxPfeNotifyGICorrupt	jnxPfeNotifyGIEEntry 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	jnxPfeNotifyGIEEntry 5	The number of notifications dropped because the notification has an illegal notification type.
jnxPfeNotifyGISample	jnxPfeNotifyGIEEntry 6	The number of sample notifications reported by the Packet Forwarding Engine controller.
jnxPfeNotifyGIGiants	jnxPfeNotifyGIEEntry 7	The number of notifications dropped because the notification is larger than the supported direct memory access (DMA) size.
jnxPfeNotifyGITtlExceeded	jnxPfeNotifyGIEEntry 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	jnxPfeNotifyGIEEntry 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	jnxPfeNotifyGIEEntry 10	The number of IP options packets sent to a Services PIC.
jnxPfeNotifyGISvcOptRe	jnxPfeNotifyGIEEntry 11	The number of IP options packets sent to the Routing Engine.
jnxPfeNotifyGIPostSvcOptOut	jnxPfeNotifyGIEEntry 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	jnxPfeNotifyGIEEntry 13	The number of TTL-expired transit packets.
jnxPfeNotifyGIDiscSample	jnxPfeNotifyGIEEntry 14	The number of sample notifications dropped because the notifications refer to discarded packets in the Packet Forwarding Engine.
jnxPfeNotifyGIRateLimited	jnxPfeNotifyGIEEntry 15	The number of notification ignored because of Packet Forwarding Engine software throttling (delaying or refusing requests).
jnxPfeNotifyGIPktGetFails	jnxPfeNotifyGIEEntry 16	The number of notifications ignored because DMA memory could not be allocated.

Table 258: 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.

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 259 on page 649](#).

Table 259: 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 260 on page 650](#).

Table 260: jnxPfeNotifyTypeEntry

Object	Object Identifier	Description
jnxPfeNotifyTypePld	jnxPfeNotifyTypeEntry1	Identifies the Packet Forwarding Engine notification type. See Table 259 on page 649 for a list of notification types.
jnxPfeNotifyTypeDescr	jnxPfeNotifyTypeEntry2	A description of the Packet Forwarding Engine notification type.
jnxPfeNotifyTypeParsed	jnxPfeNotifyTypeEntry3	The number of notifications that are parsed successfully.
jnxPfeNotifyTypeInput	jnxPfeNotifyTypeEntry4	The number of notifications whose associated packets are stored in router processor memory using direct memory access.
jnxPfeNotifyTypeFailed	jnxPfeNotifyTypeEntry5	The number of notifications that are not parsed successfully.
jnxPfeNotifyTypeIgnored	jnxPfeNotifyTypeEntry6	The number of notifications where the notification type in the message does not match any of the valid notification types.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 53

Packet Mirror MIB

- [Interpreting the Enterprise-Specific Packet Mirror MIB on page 651](#)

Interpreting the Enterprise-Specific Packet Mirror MIB

- [Packet Mirror MIB Overview on page 651](#)
- [jnxJsPacketMirrorTrapVars on page 651](#)
- [Packet Mirroring-Related Traps on page 653](#)

Packet Mirror MIB Overview

The Juniper Networks enterprise-specific Packet Mirror MIB, **jnxJsPacketMirrorMIB**, whose object ID is **{jnxJsPacketMirror 1}**, enables you to capture and view packet mirroring-related information. 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/junos13.2/topics/reference/mibs/mib-jnx-js-packet-mirror.txt.

For information about the enterprise-specific Packet Mirror MIB objects, see the following topics:

- [jnxJsPacketMirrorTrapVars on page 651](#)
- [Packet Mirroring-Related Traps on page 653](#)

jnxJsPacketMirrorTrapVars

The **jnxJsPacketMirrorTrapVars** object, whose object identifier is **{jnxJsPacketMirrorObjects 1}**, contains the objects listed in [Table 261 on page 652](#).

You can use the objects listed in [Table 261 on page 652](#) to retrieve information about the subscriber and to track their traffic statistics.

Table 261: jnxJsPacketMirrorTrapVars

Object	Object ID	Description
jnxJsPacketMirrorIdentifier	jnxJsPacketMirrorTrapVars 1	Unique identifier of the mirrored session. Consists of the version, mirror ID, and session ID.
jnxJsPacketMirrorSessionIdentifier	jnxJsPacketMirrorTrapVars 2	Accounting session ID.
jnxJsPacketMirrorTrigger	jnxJsPacketMirrorTrapVars 3	Event that triggered the packet mirroring trap.
jnxJsPacketMirrorTriggerType	jnxJsPacketMirrorTrapVars 4	Type of event that triggered the packet mirroring trap.
jnxJsPacketMirrorConfigurationSource	jnxJsPacketMirrorTrapVars 5	Source IP address from where the packet mirroring request was configured.
jnxJsPacketMirrorErrorCause	jnxJsPacketMirrorTrapVars 6	Reason for the error message generated.
jnxJsPacketMirrorErrorString	jnxJsPacketMirrorTrapVars 7	Error message as reported by the Junos OS application.
jnxJsPacketMirrorApplicationName	jnxJsPacketMirrorTrapVars 8	Name of the Junos OS application reporting the trap.
jnxJsPacketMirrorAnalyzerAddress	jnxJsPacketMirrorTrapVars 9	IP address of the analyzer.
jnxJsPacketMirrorUserName	jnxJsPacketMirrorTrapVars 10	Login ID of the user whose traffic is being mirrored.
jnxJsPacketMirrorDateAndTime	jnxJsPacketMirrorTrapVars 11	Date and time when the event occurred.
jnxJsPacketMirrorRouterId	jnxJsPacketMirrorTrapVars 12	VRF ID in Junos OS. Represented as the logical router (LR) name followed by the router instance (RI) name.
jnxJsPacketMirrorDirection	jnxJsPacketMirrorTrapVars 13	Mirroring direction. Possible values: <ul style="list-style-type: none"> • ingress (0) - Ingress traffic is mirrored. • egress (1) - Egress traffic is mirrored. • bidirectional (2) - Both ingress and egress traffic is mirrored.
jnxJsPacketMirrorTargetIpAddress	jnxJsPacketMirrorTrapVars 14	IP address of the mirrored interface.
jnxJsPacketMirrorTerminationReason	jnxJsPacketMirrorTrapVars 15	Reason for terminating a packet mirroring session.
jnxPacketMirrorCallingStationIdentifier	jnxJsPacketMirrorTrapVars 16	Unique calling station ID of the subscriber whose traffic is being monitored.
jnxPacketMirrorNasIdentifier	jnxJsPacketMirrorTrapVars 17	Unique NAS ID of the router on which the traffic is being monitored.

Table 261: jnxJsPacketMirrorTrapVars (*continued*)

Object	Object ID	Description
jnxJsPacketMirrorOctetsReceived	jnxJsPacketMirrorTrapVars 18	Number of octets of combined IPv4 and IPv6 subscriber traffic received.
jnxJsPacketMirrorOctetsTransmitted	jnxJsPacketMirrorTrapVars 19	Number of octets of combined IPv4 and IPv6 subscriber traffic transmitted.
jnxJsPacketMirrorTargetIpv6Address	jnxJsPacketMirrorTrapVars 20	IPv6 address of the mirrored interface.

Packet Mirroring-Related Traps

The **jnxJsPacketMirrorNotifications** object, whose object identifier is **{jnxJsPacketMirrorMIB 0}**, contains the objects listed in [Table 262 on page 653](#).

You can use the objects listed in [Table 262 on page 653](#) to retrieve information about the packet mirroring traps and when they are triggered.

Table 262: Packet Mirroring-Related Traps

Object	Object ID	Description
jnxJsPacketMirrorMirroringFailure	jnxJsPacketMirrorNotifications 1	The packet mirroring request failed.
jnxJsPacketMirrorLiSubscriberLoggedIn	jnxJsPacketMirrorNotifications 2	Subscriber has logged in successfully.
jnxJsPacketMirrorLiSubscriberLogInFailed	jnxJsPacketMirrorNotifications 3	Subscriber was unable to log in.
jnxJsPacketMirrorLiSubscriberLoggedOut	jnxJsPacketMirrorNotifications 4	Subscriber has logged out.
jnxJsPacketMirrorLiServiceActivated	jnxJsPacketMirrorNotifications 5	Packet mirroring on an interface is now active.
jnxJsPacketMirrorLiServiceActivationFailed	jnxJsPacketMirrorNotifications 6	Packet mirroring on an interface has failed.
jnxJsPacketMirrorLiServiceDeactivated	jnxJsPacketMirrorNotifications 7	Packet mirroring on an interface is now deactivated.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

PAE Extension MIB

- [Interpreting the Enterprise-Specific PAE Extension MIB on page 655](#)

Interpreting the Enterprise-Specific PAE Extension MIB

- [PAE Extension MIB on page 655](#)
- [jnxAuthProfileName on page 655](#)
- [jnxPaeAuthConfigTable on page 656](#)
- [jnxStaticMacAuthBypassTable on page 656](#)
- [jnxStaticMacAuthBypassIfTable on page 657](#)

PAE Extension MIB

The Juniper Networks enterprise-specific Port Access Entity (PAE) Extension MIB, whose object identifier is **{jnxExSwitching 3}**, is an extension of the standard IEEE802.1x PAE Extension MIB, and contains information for static MAC authentication. The enterprise-specific PAE Extension MIB has two branches, **jnxPaeExtensionMIBNotification** and **jnxPaeExtensionMIBObjects**.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pae-extension.txt.

For information about the enterprise-specific PAE Extension MIB objects, see the following topics:

- [jnxAuthProfileName on page 655](#)
- [jnxPaeAuthConfigTable on page 656](#)
- [jnxStaticMacAuthBypassTable on page 656](#)
- [jnxStaticMacAuthBypassIfTable on page 657](#)

jnxAuthProfileName

The **jnxAuthProfileName** object, whose object identifier is **{jnxPaeExtensionMIBObjects 1}**, contains the authentication profile name. The authentication profile contains the RADIUS server IP address, the port number, and the secret key.

jnxPaeAuthConfigTable

The **jnxPaeAuthConfigTable**, whose object identifier is **{jnxPaeExtensionMIBObjects 2}**, contains the configuration objects for the Authenticator PAE associated with each port. Each **jnxPaeAuthConfigEntry**, indexed with **dot1xPaePortNumber** from the standard IEEE802.1x PAE Extension MIB, contains the objects listed in [Table 263 on page 656](#).

Table 263: jnxPaeAuthConfigTable

Object	Object ID	Description
jnxPaeAuthConfigMacAuthStatus	jnxPaeAuthConfigEntry 1	Shows whether MAC authentication is enabled on the specified PAE port.
jnxPaeAuthConfigGuestVlan	jnxPaeAuthConfigEntry 2	Specifies the VLAN to which an unauthenticated client moves. This object can contain a string of not more than 255 characters.
jnxPaeAuthConfigNumberRetries	jnxPaeAuthConfigEntry 3	Specifies the maximum number of failed authentication retries allowed on an interface before the interface transitions into the quiet period. No authentication happens on the interface during the quiet period.

jnxStaticMacAuthBypassTable

The **jnxStaticMacAuthBypassTable**, whose object identifier is **{jnxPaeExtensionMIBObjects 3}**, contains a static list of MAC addresses specified by a user. The static MAC address list contains the MAC addresses of clients associated with a port. The clients whose MAC addresses are in the MAC address list are allowed to connect to the port without authentication. The 802.1X or MAC authentication process is initiated for a connection request only when a matching entry is not available for the client in the **jnxStaticMacAuthBypassTable**.

The **jnxStaticMacAuthBypassTable** allows devices like printers that do not support 802.1X to connect to 802.1X-enabled ports.

Each **jnxStaticMacAuthBypassEntry**, whose object identifier is **{jnxStaticMacAuthBypassTable 1}**, contains the objects listed in [Table 264 on page 656](#).

Table 264: jnxStaticMacAuthBypassTable

Object	Object ID	Description
jnxStaticMacAddress	jnxStaticMacAuthBypassEntry 1	Specifies the MAC address of the client connected to the PAE port.
jnxStaticMacVlanName	jnxStaticMacAuthBypassEntry 2	Specifies the VLAN to which the client is assigned.

jnxStaticMacAuthBypassIfTable

The `jnxStaticMacAuthBypassIfTable`, whose object identifier is `{jnxPaeExtensionMIBObjects 4}`, contains a list of interfaces associated with the MAC addresses in the `jnxStaticMacAuthBypassTable`.

Each `jnxStaticMacAuthBypassIfEntry`, whose object identifier is `{jnxStaticMacAuthBypassIfTable 1}`, is indexed with `jnxStaticMacAddress` (from `jnxStaticMacAuthBypassTable`) and `jnxStaticMacIfIndex`. The `jnxStaticMacIfIndex` contains a list of interfaces from which a MAC address is allowed. If the interface associated with a MAC address does not match the one stored in this entry, the authentication bypass does not occur.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Passive Monitoring MIB

- Interpreting the Enterprise-Specific Passive Monitoring MIB on page 659

Interpreting the Enterprise-Specific Passive Monitoring MIB

- Passive Monitoring MIB on page 659
- jnxPMonFlowTable on page 660

Passive Monitoring MIB

The Juniper Networks enterprise-specific Passive Monitoring MIB, whose object identifier is **{jnxMibs 19}**, performs traffic-flow monitoring and lawful interception of packets transiting between two routers. This MIB enables you to do the following:

- Gather and export detailed information about IPv4 and IPv6 traffic flows between source and destination nodes in your network.
- Sample all incoming IPv4 and IPv6 traffic on the monitoring interface and present the data in **cflowd** record format.
- Encrypt or tunnel outgoing **cflowd** records, intercepted IPv4 and IPv6 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.

The Passive Monitoring MIB has three tables: **jnxPMonFlowTable**, **JnxPMonErrorTable**, and **jnxPMonMemoryTable**. **jnxPMonFlowTable** monitors and collects statistics on the flow of traffic on a Passive Monitoring PIC. **jnxPMonErrorTable** monitors and collects statistics on packet and memory errors on a Passive Monitoring PIC. **jnxPMonMemoryTable** monitors and collects statistics on memory usage on a Passive Monitoring PIC. For information about system requirements, see the *Junos OS Feature Guide*. For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pmon.txt.

For information about the Passive Monitoring MIB objects, see the following topic:

- [jnxPMonFlowTable](#) on page 660

jnxPMonFlowTable

jnxPMonFlowTable has an object identifier of **{jnxPMon 1}**. Its entries are represented by **jnxPMonFlowEntry**, which contains the objects listed in [Table 265 on page 660](#).

Table 265: jnxPMFlowEntry

Object	Object Identifier	Description
jnxPMonCurrentActiveFlows	jnxPMonFlowEntry 1	Monitors the number of currently active flows on a Passive Monitoring PIC.
jnxPMonTotalFlows	jnxPMonFlowEntry 2	Monitors the total flows on a Passive Monitoring PIC.
jnxPMonTotalFlowsPackets	jnxPMonFlowEntry 3	Monitors the total packet flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowsPackets	jnxPMonFlowEntry 4	Monitors the number of packets in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsBytes	jnxPMonFlowEntry 5	Monitors the number of total of bytes in all flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowBytes	jnxPMonFlowEntry 6	Monitors the number of bytes in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsExpired	jnxPMonFlowEntry 7	Monitors the number of total flows expired on a Passive Monitoring PIC.
jnxPMonTotalFlowsAged	jnxPMonFlowEntry 8	Monitors the number of total flows aged on a Passive Monitoring PIC.
jnxPMonTotalFlowsExported	jnxPMonFlowEntry 9	Monitors the number of total flows exported on a Passive Monitoring PIC.
jnxPMonTotalFlowsPacketsExported	jnxPMonFlowEntry 10	Monitors the number of total flow packets exported on a Passive Monitoring PIC.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)

- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 56

PING MIB

- [Interpreting the Enterprise-Specific Ping MIB on page 663](#)

Interpreting the Enterprise-Specific Ping MIB

- [PING MIB on page 663](#)
- [jnxPingCtlTable on page 664](#)
- [jnxPingResultsTable on page 668](#)
- [jnxPingProbeHistoryTable on page 671](#)
- [jnxPingLastTestResultTable on page 673](#)

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/junos13.2/topics/reference/mibs/mib-jnx-ping.txt.

For information about using the Ping MIB and enterprise-specific Ping MIB, see the *Network Management Administration Guide for Routing Devices*.

For information about the enterprise-specific PING MIB objects, see the following topics:

- [jnxPingCtlTable on page 664](#)
- [jnxPingResultsTable on page 668](#)
- [jnxPingProbeHistoryTable on page 671](#)
- [jnxPingLastTestResultTable on page 673](#)

jnxPingCtlTable

The enterprise-specific Ping MIB structure includes one main object, **jnxPingCtlTable**, whose object identifier is **{jnxPingObjects 2}**, and defines the **jnxPing** control table for providing enterprise-specific options to the corresponding **pingCtlEntry**. **jnxPingCtlTable** monitors thresholds; for example, the maximum allowed jitter in the trip time during a test.

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 266 on page 664](#).

Table 266: 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: <ul style="list-style-type: none"> • pingCtlIfIndex (see pingCtlTable in "PING MIB" on page 663) • 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.

Table 266: 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.

Table 266: 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.

Table 266: jnxPingCtlTable (*continued*)

Object	Object Identifier	Description
jnxPingCtlTrapGeneration	jnxPingCtlEntry 16	<p>The value of this object determines when and if to generate a notification for this entry.</p> <p>rttThreshold(0)—Generate a jnxPingRttThresholdExceeded notification when the configured RTT threshold is exceeded.</p> <p>rttStdDevThreshold(1)—Generate a jnxPingRttStdDevThresholdExceeded notification when the configured RTT standard deviation threshold is exceeded.</p> <p>rttJitterThreshold(2)—Generate a jnxPingRttJitterThresholdExceeded notification when the configured RTT jitter threshold is exceeded.</p> <p>egressThreshold(3)—Generate a jnxPingEgressThresholdExceeded notification when the configured egress threshold is exceeded. This applies only if the probe type supports one-way measurements.</p>

Table 266: 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.

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 267 on page 668](#).

Table 267: jnxPingsResultsEntry

Object	Object Identifier	Description
jnxPingResultsRttUs	jnxPingResultsEntry 1	The round-trip delays measured for the most recent successful probe during this test, in microseconds.

Table 267: jnxPingsResultsEntry (continued)

Object	Object Identifier	Description
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.
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.

Table 267: jnxPingsResultsEntry (continued)

Object	Object Identifier	Description
jnxPingResultsStddevEgressUs	jnxPingResultsEntry 11	The standard deviation of the egress trip delays measured over all probes during this test, in microseconds. This applies only if the probe type (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.
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.

Table 267: jnxPingsResultsEntry (continued)

Object	Object Identifier	Description
jnxPingResultsJitterEgressUs	jnxPingResultsEntry 18	The jitter of the egress trip delays measured for all probes during this test, in microseconds. This applies only if the probe type (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.

jnxPingProbeHistoryTable

jnxpingProbeHistoryTable, whose object identifier is **{jnxPingObjects 4}**, contains the history of all ping tests.

The **jnxPingProbeHistoryEntry** objects are listed in [Table 268 on page 672](#).

Table 268: 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.
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.

Table 268: jnxPingProbeHistoryEntry (*continued*)

Object	Object Identifier	Description
jnxPingProbeHistoryEgressJitterUs	jnxPingProbeHistoryEntry 5	The time difference, in microseconds, between the maximum and minimum egress trip times. Each history entry provides a running calculation of the jitter (calculated over the current test) at the time a probe was completed. This applies only if the probe type (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.

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 269 on page 673](#).

Table 269: jnxPingLastTestResultEntry

Object	Object Identifier	Description
jnxPingLastTestResultProbeResponses	jnxPingLastTestResultEntry 1	The number of responses received in the most recently completed test.

Table 269: jnxPingLastTestResultEntry (*continued*)

Object	Object Identifier	Description
jnxPingLastTestResultSentProbes	jnxPingLastTestResultEntry 2	The number of probes sent in the most recently completed test.
jnxPingLastTestResultSumRttUs	jnxPingLastTestResultEntry 3	The sum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinRttUs	jnxPingLastTestResultEntry 4	The minimum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMaxRttUs	jnxPingLastTestResultEntry 5	The maximum of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultAvgRttUs	jnxPingLastTestResultEntry 6	The average of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultStdDevRttUs	jnxPingLastTestResultEntry 7	The standard deviation of the round-trip delays measured for all the probes during the most recently completed test, in microseconds.
jnxPingLastTestResultMinEgressUs	jnxPingLastTestResultEntry 8	The minimum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (pingCtlType) provides one-way delay measurements. For all other probe types, their values are irrelevant and this object returns 0.

Table 269: jnxPingLastTestResultEntry (*continued*)

Object	Object Identifier	Description
jnxPingLastTestResultMaxEgressUs	jnxPingLastTestResultEntry 9	The maximum of the egress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (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.
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.

Table 269: jnxPingLastTestResultEntry (*continued*)

Object	Object Identifier	Description
jnxPingLastTestResultMaxIngressUs	jnxPingLastTestResultEntry 13	The maximum of the ingress trip delays measured over all probes during the most recently completed test, in microseconds. This applies only if the probe type (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.
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.
jnxPingLastTestResultPeakToPeakJitterRttUs	jnxPingLastTestResultEntry 16	The difference between the minimum and maximum delays over the course of the last completed test, in microseconds.

Table 269: jnxPingLastTestResultEntry (*continued*)

Object	Object Identifier	Description
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.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 57

Policy Objects MIB

- [Interpreting the Enterprise-Specific Policy Objects MIB on page 679](#)

Interpreting the Enterprise-Specific Policy Objects MIB

- [Policy Objects MIB on page 679](#)
- [Security Policy Table on page 680](#)
- [jnxJsPolicyStatsTable on page 681](#)

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/junos13.2/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 680](#)
- [jnxJsPolicyStatsTable on page 681](#)

Security Policy Table

Table 270 on page 680 describes the objects in the Security Policy Table.

Table 270: Security Policy Table

Object	Object ID	Description
<code>jnxJsPolicyNumber</code>	<code>jnxJsPolicyObjects 1</code>	Number of policies (regardless of their current state) present on this system.
<code>jnxJsPolicyTable</code>	<code>jnxJsPolicyObjects 2</code>	<p>Exposes the security policy entries. Security devices and routers provide a network boundary with a single point of entry and exit, which allows the screening and directing of traffic through the implementation of access policies. The access policies can permit, deny, encrypt, authenticate, prioritize, schedule, and monitor the traffic flow through the firewall.</p> <p>This table lists entries of policy. The number of policies are indicated by <code>jnxJsPolicyNumber</code>.</p>
<code>jnxJsPolicyEntry</code>	<code>jnxJsPolicyTable 1</code>	<p>An entry contains a security policy.</p> <p>Indexes:</p> <ul style="list-style-type: none"> • <code>nxJsPolicyFromZone</code> • <code>jnxJsPolicyToZone</code> • <code>jnxJsPolicyName</code> <p>Security policies are configured under FromZone and ToZone directions. Under a specific zone direction, each security policy contains a name, match-criteria, action, and other options.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • <code>jnxJsPolicyFromZone</code> • <code>jnxJsPolicyToZone</code> • <code>jnxJsPolicyName</code> • <code>jnxJsPolicySequenceNumber</code> • <code>jnxJsPolicyAction</code> • <code>jnxJsPolicyScheduler</code> • <code>jnxJsPolicyState</code> • <code>jnxJsPolicyStatsAvailability</code> • <code>jnxJsPolicyPerSecBytesThreshold</code> • <code>jnxJsPolicyPerMinKbytesThreshold</code>
<code>jnxJsPolicyFromZone</code>	<code>jnxJsPolicyEntry 1</code>	FromZone name.

Table 270: Security Policy Table (*continued*)

Object	Object ID	Description
<code>jnxJsPolicyToZone</code>	<code>jnxJsPolicyEntry 2</code>	ToZone name.
<code>jnxJsPolicyName</code>	<code>jnxJsPolicyEntry 3</code>	Name of the policy defined. The name consists of up to 256 ASCII characters and uniquely identifies the policy entry.
<code>jnxJsPolicySequenceNumber</code>	<code>jnxJsPolicyEntry 4</code>	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.
<code>jnxJsPolicyAction</code>	<code>jnxJsPolicyEntry 5</code>	Indication of the actions performed when the criteria are matched. The actions permit , reject , and deny are user-configured policies.
<code>jnxJsPolicyScheduler</code>	<code>jnxJsPolicyEntry 6</code>	Name of the schedule attached to this policy. Certain schedules have a specified duration that can affect the status of the policy.
<code>jnxJsPolicyState</code>	<code>jnxJsPolicyEntry 7</code>	State of this policy: active, inactive, or unavailable. The state can be affected by the scheduler if the scheduler has a specified duration.
<code>jnxJsPolicyStatsAvailability</code>	<code>jnxJsPolicyEntry 8</code>	Indication of whether the statistics counters are available and are actively updated. If available, a matching <code>jnxJsPolicyStatsEntry</code> exists for the policy.
<code>jnxJsPolicyPerSecBytesThreshold</code>	<code>jnxJsPolicyEntry 9</code>	Indication of the threshold value of bytes per second
<code>jnxJsPolicyPerMinKbytesThreshold</code>	<code>jnxJsPolicyEntry 10</code>	Indication of the threshold value of KB per minute

`jnxJsPolicyStatsTable`

`jnxJsPolicyStatsTable`, whose object ID is `{jnxJsPolicyObjects 3}`, exposes the security policy statistics entries listed in [Table 271 on page 682](#). These statistics can be enabled and disabled by configuration on a per-policy basis.

Table 271: jnxJsPolicyStatsTable

Object	Object ID	Description
jnxJsPolicyStatsEntry	jnxJsPolicyStatsTable 1	<p>Contains security policy statistics.</p> <p>Indexes:</p> <ul style="list-style-type: none"> • jnxJsPolicyFromZone • jnxJsPolicyToZone • jnxJsPolicyName <p>Security policies are configured under FromZone and ToZone direction. Under a specific zone direction, each security policy contains name, match-criteria, action, and other options.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • 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.

Table 271: 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**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 58

Power Supply Unit MIB

- Interpreting the Enterprise-Specific Power Supply Unit MIB on page 685

Interpreting the Enterprise-Specific Power Supply Unit MIB

- Power Supply Unit MIB on page 685
- Power Supply Unit Environment Group on page 685
- Power Supply Unit Device Power Group on page 686
- Power Supply Unit Outlets Group on page 686
- Power Source Equipment Objects Group on page 687
- Power Supply Unit FPC Power Allocated Information on page 688

Power Supply Unit MIB

The **jnxPsuMIB** MIB module is for enabling power monitoring and management of a switch. This is handled by two modules: **jnxPsuNotifications** module, whose object identifier is **{jnxPsuMIB1}**, and **jnxPsuObjects** module, whose object identifier is **{jnxPsuMIB2}**.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-power-supply-unit.txt.

For information about the Juniper Networks enterprise-specific Power Supply Unit (PSU) MIB objects, see the following topics:

- Power Supply Unit Environment Group on page 685
- Power Supply Unit Device Power Group on page 686
- Power Supply Unit Outlets Group on page 686
- Power Source Equipment Objects Group on page 687
- Power Supply Unit FPC Power Allocated Information on page 688

Power Supply Unit Environment Group

The **jnxPsuEnvironmentTable** lists all the environment entries for each PSU component. The **jnxPsuEnvironmentEntry** contains the objects listed in [Table 272 on page 686](#).

Table 272: jnxPsuEnvironmentTable

Object	Object Identifier	Description
jnxPsuThermalValue	jnxPsuEnvironmentEntry 1	The temperature (in degrees Celsius) for each PSU component.
jnxPsuHumidityValue	jnxPsuEnvironmentEntry 2	The humidity (in percentage) for each PSU component.

Power Supply Unit Device Power Group

The **jnxPsuTable** lists all the power entries for each PSU component. The **jnxPsuEntry** contains the objects listed in [Table 273 on page 686](#).

Table 273: jnxPsuTable

Object	Object Identifier	Description
jnxPsuAvgPower	jnxPsuEntry 1	The buffer that specifies the average power in watts used for each component.
jnxPsuMaxPower	jnxPsuEntry 2	The buffer that specifies the maximum power in watts available for each component.
jnxPsuMode	jnxPsuEntry 3	The mode for each PSU component.
jnxPsuOutletCount	jnxPsuEntry 4	The number of outlets present on a particular PSU component. The default number is 0.

Power Supply Unit Outlets Group

The **jnxPsuOutletTable** lists all the outlet entries. The number of entries is specified by the **jnxPsuOutletPsuIndex**. The **jnxPsuOutletEntry** contains the objects listed in [Table 274 on page 686](#).

Table 274: jnxPsuOutletTable

Object	Object Identifier	Description
jnxPsuOutletName	jnxPsuOutletEntry 1	The outlet name associated with the power supply unit for each PSU Component.
jnxPsuOutletDescription	jnxPsuOutletEntry 2	The outlet description associated with the power supply unit for each PSU Component.
jnxPsuOutletAvgPower	jnxPsuOutletEntry 3	The buffer that specifies the average power in watts used for each component.
jnxPsuOutletMaxPower	jnxPsuOutletEntry 4	The buffer that specifies the maximum power in watts available for each component.

Table 274: jnxPsuOutletTable (*continued*)

Object	Object Identifier	Description
jnxPsuOutletCurrent	jnxPsuOutletEntry 5	The actual current in milliamps that flows through the given outlet.
jnxPsuOutletStatus	jnxPsuOutletEntry 8	The operational status for the given outlet. This can also be used to set the outlet state.

Power Source Equipment Objects Group

The **pethObjects** module consists of the following:

- The **pethPsePortTable** lists the objects that display and control the power characteristics of Power over Ethernet (POE) ports on a PSE device. The objects are identified by **pethPsePortEntry**.
- The **pethPsePortGroupIndex** identifies the group containing the port to which a power Ethernet PSE is connected.
- The **pethPsePortIndex** specifies the PSE port within the group **pethPsePortGroupIndex** to which a PSE entry is connected.
- The **pethPsePortAdminEnable** specifies whether or not an interface can provide the PSE functions.
- The **pethPsePortPowerPairsControlAbility** specifies whether or not the device can control power pairs in order to switch pins for sourcing power.
- The **pethPsePortPowerPairs** specifies power controls for the power pairs in use.
 - **pethPsePortPowerPairsControl** value is true - Object is writable
 - **pethPsePortPowerPairsControl** value is signal(1) - Signal pairs only are in use
 - **pethPsePortPowerPairsControl** value is spare(2) - Spare pairs only are in use
- The **pethPsePortDetectionStatus** specifies the operational status of the port PD detection.
- The **pethPsePortPowerPriority** specifies the priority of the port to aid efficient power management.
- The **pethPsePortMPSAbsentCounter** displays the PSE state diagram transitions from **POWER_ON** to **IDLE** to **tmpdo_timer_done**.
- The **pethPsePortType** specifies the type of powered device that is connected to the port. The default value is a zero-length octet string.
- The **pethPsePortPowerClassifications** specifies how the different terminals on the Power over LAN network are tagged according to their power consumption.
- The **pethPsePortInvalidSignatureCounter** displays the invalid state of the port when the port enters the state **SIGNATURE_INVALID**.

- The **pethPsePortPowerDeniedCounter** indicates that the port is denied power when the port enters the state **POWER_DENIED**.
- The **pethPsePortOverLoadCounter** indicates that the port is overloaded with power when the port enters the state **ERROR_DELAY_OVER**.
- The **pethPsePortShortCounter** indicates that the port is not supplied with sufficient power when the port enters the state **ERROR_DELAY_OVER**.

Power Supply Unit FPC Power Allocated Information

The **jnxPsuFpcPowerTable** lists the entries for each FPC (Flexible PIC Concentrator), their assigned priority levels, and allocated power. The **jnxPsuFpcPowerEntry** contains the objects listed in [Table 275 on page 688](#).

Table 275: jnxPsuFpcPowerTable

Object	Object Identifier	Description
jnxPsuFpcPowerPriority	jnxPsuFpcPowerEntry 1	The power budget priority assigned to the FPC. A lower number indicates a higher priority.
jnxPsuFpcPowerAllocated	jnxPsuFpcPowerEntry 2	The power (in Watts) allocated to the FPC.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 59

PPP MIB

- Interpreting the Enterprise-Specific PPP MIB on page 689

Interpreting the Enterprise-Specific PPP MIB

- PPP MIB on page 689
- PPP MIB Textual Conventions on page 690
- Managed Objects for PPP Link Control Protocol Layer on page 691
- Managed Objects for PPP IP Network Control Protocol Layer on page 696
- Managed Objects for PPP OSI Network Control Protocol Layer on page 698
- Managed Objects for PPP Session Attributes on page 701
- Managed Objects for Multilink PPP Layer on page 703
- Managed Objects for PPP Interface Summary Counts on page 708
- Managed Objects for PPP IPv6 Network Control Protocol Layer on page 713
- Managed Objects for PPP Global Attributes on page 715

PPP MIB

The Juniper Networks enterprise-specific PPP MIB, whose object ID is `{jnxPppMibRoot 1}`, extends SNMP support for the Point-to-Point Protocol (PPP) modules in Junos OS.

The PPP MIB stores PPP-related information, such as the authentication type, status of the operation of network protocols over each link, PPP session status, PPP interface characteristics, and statistics. You can view this information by using SNMP **get** and **get-next** requests.



NOTE: The PPP MIB is supported on the Common Edge PPP process, `jpppd`. SNMP support for PPP over PPPoE interfaces is active only if the `jpppd` process is enabled.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ppp.txt.

For information about enterprise-specific PPP MIB objects, see the following topics:

- [PPP MIB Textual Conventions on page 690](#)
- [Managed Objects for PPP Link Control Protocol Layer on page 691](#)
- [Managed Objects for PPP IP Network Control Protocol Layer on page 696](#)
- [Managed Objects for PPP OSI Network Control Protocol Layer on page 698](#)
- [Managed Objects for PPP Session Attributes on page 701](#)
- [Managed Objects for Multilink PPP Layer on page 703](#)
- [Managed Objects for PPP Interface Summary Counts on page 708](#)
- [Managed Objects for PPP IPv6 Network Control Protocol Layer on page 713](#)
- [Managed Objects for PPP Global Attributes on page 715](#)

PPP MIB Textual Conventions

[Table 276 on page 690](#) contains information about the textual conventions used in the enterprise-specific PPP MIB.

Table 276: PPP MIB Textual Conventions

Object	Description	Syntax
jnxPppAuthentication	Specifies the types of PPP authentication used.	<p>This object uses the following integer values:</p> <ul style="list-style-type: none"> • 0—None (no authentication is negotiated) • 1—pap (Password Authentication Protocol negotiation only) • 2—chap (Challenge Handshake Authentication Protocol negotiation only) • 3—papChap (PAP negotiation is attempted first; if that fails, attempts CHAP negotiation) • 4—chapPap (CHAP negotiation is attempted first; if that fails, attempts PAP negotiation)
jnxPppMlPppBundleName	Specifies the name of the MLPPP bundle.	An octet string that contains the name of the MLPPP bundle.
jnxPppAuthentication2	Specifies the type of PPP authentication used.	<p>This object uses the following integer values:</p> <ul style="list-style-type: none"> • 0—None (no authentication is negotiated) • 1—pap (PAP negotiation only) • 2—chap (CHAP negotiation only) • 3—eap (Extensible Authentication Protocol negotiation only)
jnxNibbleConfig	Specifies the configuration variable that consists of nibbles (4 bits).	-

Managed Objects for PPP Link Control Protocol Layer

The **jnxPppLcp** object, whose object identifier is **{jnxPPPObjects 1}**, contains the objects listed in **jnxPppLinkStatusTable**, **jnxPppLinkConfigTable**, and **jnxPppNextIfIndex**. You can use the managed objects listed in these tables to retrieve information about the PPP interfaces present in the system.

For information about **jnxPppLinkStatusTable**, **jnxPppLinkConfigTable**, and **jnxPppNextIfIndex**, see the following sections:

- [jnxPppLinkStatusTable on page 691](#)
- [jnxPppLinkConfigTable on page 694](#)
- [jnxPppNextIfIndex on page 696](#)

jnxPppLinkStatusTable

jnxPppLinkStatusTable, whose object identifier is **{jnxPppLcp 1}**, contains the objects listed in [Table 277 on page 692](#) and provides information about the characteristics of the PPP interface.

Each **jnxPppLinkStatusEntry**, whose object identifier is **{jnxPppLinkStatusTable 1}**, maps to a specific characteristic of the PPP interface present in the system.

Table 277: jnxPppLinkStatusTable

Object	Object ID	Description
jnxPppLinkStatusTerminateReason	jnxPppLinkStatusEntry 1	<p>Reason the PPP link was terminated. Possible values:</p> <ul style="list-style-type: none"> • none (0) - None. • other (1) - Not specified. • adminDisable (2) - Interface was administratively disabled. • lowerlayerDown (3) - Underlying interface is down. • noUpperInterface (4) - No interface above PPP. • authenticationFailure (5) - Authentication failed. • peerTerminated (6) - Peer initiated termination. • peerRenegotiated (7) - Peer initiated renegotiation. • maxretriesExceeded (8) - Maximum number of retries exceeded. • negotiationFailure (9) - Failed to negotiate LCP option. • keepaliveFailure (10) - Keepalive failed. • sessionTimeout (11) - Maximum session period expired. • inactivityTimeout (12) - Maximum inactivity period expired. • addressLeaseExpired (13) - Lease for network address expired. • adminLogout (14) - Session administratively terminated. • tunnelFailed (15) - Associated tunnel failed. • tunnelDisconnected (16) - Associated tunnel disconnected. • loopback (17) - Loopback detected.

Table 277: jnxPppLinkStatusTable (*continued*)

Object	Object ID	Description
jnxPppLinkStatusTerminateNegFailOption	jnxPppLinkStatusEntry 2	<p>PPP LCP option for which the negotiation failed when jnxPppLinkStatusTerminateReason has a value of negotiationFailure (9). Possible values:</p> <ul style="list-style-type: none"> • none (0) - None. • other (1) - Not specified. • localMru (2) - MRU setting of the local device. • remoteMru (3) - MRU setting of the remote device. • localMagicNumber (4) - Configuration option that provides a method to detect looped-back links on the local device. • remoteMagicNumber (5) - Configuration option that provides a method to detect looped-back links on the remote device. • localAuthentication (6) - Authentication information about the local entity. • localToRemoteProtocolCompression (7) - Protocol compression is enabled or disabled. • localToRemoteACCompression (8) - AC compression is enabled or disabled.
jnxPppLinkStatusInKeepaliveRequests	jnxPppLinkStatusEntry 3	Number of keepalive requests received.
jnxPppLinkStatusOutKeepaliveRequests	jnxPppLinkStatusEntry 4	Number of keepalive requests transmitted.
jnxPppLinkStatusInKeepaliveReplies	jnxPppLinkStatusEntry 5	Number of keepalive replies received.
jnxPppLinkStatusOutKeepaliveReplies	jnxPppLinkStatusEntry 6	Number of keepalive replies transmitted.
jnxPppLinkStatusKeepaliveFailures	jnxPppLinkStatusEntry 7	Number of keepalive failures detected.
jnxPppLinkStatusLocalMagicNumber	jnxPppLinkStatusEntry 8	Magic number negotiated for the local side.
jnxPppLinkStatusRemoteMagicNumber	jnxPppLinkStatusEntry 9	Magic number negotiated for the remote side.
jnxPppLinkStatusLocalAuthentication	jnxPppLinkStatusEntry 10	Authentication protocol negotiated for the local side.

Table 277: jnxPppLinkStatusTable (*continued*)

Object	Object ID	Description
jnxPppLinkStatusTunnellfIndex	jnxPppLinkStatusEntry 11	IfIndex of an associated interface pertaining to a tunneling protocol. Zero indicates that no such interface exists.
jnxPppLinkStatusRenegoTerminates	jnxPppLinkStatusEntry 12	Number of times link was terminated because peer exceeded the maximum renegotiation attempts.

jnxPppLinkConfigTable

jnxPppLinkConfigTable, whose object identifier is **{jnxPppLcp 2}**, contains the objects listed in [Table 278 on page 694](#) and provides information about the PPP interface.

Each **jnxPppLinkConfigEntry**, whose object identifier is **{jnxPppLinkConfigTable 1}**, maps to a specific characteristic of the PPP interface present in the system.

Table 278: jnxPppLinkConfigTable

Object	Object ID	Description
jnxPppLinkConfigIfIndex	jnxPppLinkConfigEntry 1	IfIndex of the PPP interface.
jnxPppLinkConfigRowStatus	jnxPppLinkConfigEntry 2	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy
jnxPppLinkConfigLowerIfIndex	jnxPppLinkConfigEntry 3	IfIndex of the interface over which this PPP interface is layered. Zero indicates no layering.
jnxPppLinkConfigKeepalive	jnxPppLinkConfigEntry 4	Keepalive duration, in seconds. Value of zero disables keepalive.
jnxPppLinkConfigAuthentication	jnxPppLinkConfigEntry 5	Type of authentication, to be negotiated with a peer. Possible values: <ul style="list-style-type: none"> • none - No authentication is negotiated. • pap - PAP negotiation only. • chap - CHAP negotiation only. • papChap - PAP negotiation is attempted first; if that fails, CHAP negotiation is attempted. • chapPap - CHAP negotiation is attempted first; if that fails, PAP negotiation is attempted.
jnxPppLinkConfigMaxAuthenRetries	jnxPppLinkConfigEntry 6	Number of authentication retries permitted, in addition to a failed initial attempt.

Table 278: jnxPppLinkConfigTable (*continued*)

Object	Object ID	Description
jnxPppLinkConfigStandardIfIndex	jnxPppLinkConfigEntry 7	IfIndex value for this interface in the standard PPP MIBs.
jnxPppLinkConfigChapMinChallengeLength	jnxPppLinkConfigEntry 8	Minimum value of the CHAP authenticator challenge length value.
jnxPppLinkConfigChapMaxChallengeLength	jnxPppLinkConfigEntry 9	Maximum value of the CHAP authenticator challenge length value.
jnxPppLinkConfigPassiveMode	jnxPppLinkConfigEntry 10	Status of the LCP state machine. Possible values: <ul style="list-style-type: none"> enabled - Status of the LCP state machine is in passive mode. disabled - Status of the LCP state machine is not in a passive mode.
jnxPppLinkConfigAuthenticatorLogicalSystem	jnxPppLinkConfigEntry 11	Name of the logical system used for authentication on the PPP interface.
jnxPppLinkConfigAuthenticatorRoutingInstance	jnxPppLinkConfigEntry 12	Name of the routing instance used for authentication on the PPP interface.
jnxPppLinkConfigAaaProfile	jnxPppLinkConfigEntry 13	Name of the access profile used for authentication on the PPP interface.
jnxPppLinkConfigAuthentication2	jnxPppLinkConfigEntry 14	Type of authentication. Possible values: <ul style="list-style-type: none"> none (0) pap (1) chap (2) eap (3)
jnxPppLinkConfigIgnoreMagicNumberMismatch	jnxPppLinkConfigEntry 15	Action to be taken when the peer has not negotiated yet or has sent either null or invalid magic number in the LCP packet. Possible values: <ul style="list-style-type: none"> Ignore the mismatch and retain the connection (1) Disallow the match and terminate the connection (2)
jnxPppLinkConfigMaxLcpRenegotiation	jnxPppLinkConfigEntry 16	Maximum number of allowed lcp renegotiation attempts from peer. NOTE: This attribute is not supported. The attribute returns the default value as mentioned in the description.

jnxPppNextIfIndex

The **jnxPppNextIfIndex** object, whose object identifier is **{jnxPppLcp 3}**, contains the object listed in [Table 279 on page 696](#). Successive **Get** requests usually return different values, to avoid collisions among clients seeking to create table entries simultaneously.

Table 279: jnxPppNextIfIndex

Object	Object ID	Description
jnxPppNextIfIndex	jnxPppLcp 3	Coordinates allocation of ifIndex value for entries in jnxPppLinkConfigTable .

Managed Objects for PPP IP Network Control Protocol Layer

jnxPppIpTable and **jnxPppIpConfigTable**, whose object identifier is **{jnxPPPObjects 3}**, contains the objects listed in [Table 280 on page 696](#) and [Table 281 on page 698](#), respectively. You can use these managed objects to retrieve information about the IPCP status information for a specific PPP link.

For information about **jnxPppIpTable** and **jnxPppIpConfigTable**, see the following sections:

- [jnxPppIpTable on page 696](#)
- [jnxPppIpConfigTable on page 697](#)

jnxPppIpTable

jnxPppIpTable, whose object identifier is **{jnxPppIp 1}**, contains the objects listed in [Table 280 on page 696](#) and provides IPCP status information for a particular PPP link.

Each **jnxPppIpEntry**, whose object identifier is **{jnxPppIpTable 1}**, maps to a specific IP parameter for the local PPP entity.

Table 280: jnxPppIpTable

Object	Object ID	Description
jnxPppIpServiceStatus	jnxPppIpEntry 1	Indication as to whether IP protocol service is operating over this PPP link.

Table 280: jnxPppIpTable (continued)

Object	Object ID	Description
jnxPppIpTerminateReason	jnxPppIpEntry 2	Reason the IPCP link was terminated. Possible values: <ul style="list-style-type: none"> • none (0) - None. • other (1) - Not specified. • noService (2) - No IP service was configured on this PPP link. • admin (3) - Administratively disabled. • linkDown (4) - Underlying link is down. • peerTerminated (5) - Peer initiated termination. • peerRenegotiated (6) - Peer initiated renegotiation. • maxRetriesExceeded (7) - Configured maximum number of retries exceeded. • negotiationFailure (8) - Failed to negotiate IPCP option.
jnxPppIpTerminateNegfailOption	jnxPppIpEntry 3	PPP IPCP option for which the negotiation failed when jnxPppIpTerminateReason has a value of negotiationFailure(8).
jnxPppIpLocalIpAddress	jnxPppIpEntry 4	IP address used by the local side.
jnxPppIpRemoteIpAddress	jnxPppIpEntry 5	IP address used by the remote side.
jnxPppIpRemotePrimaryDnsAddress	jnxPppIpEntry 6	Primary DNS server used by the remote side.
jnxPppIpRemoteSecondaryDnsAddress	jnxPppIpEntry 7	Secondary DNS server used by the remote side.
jnxPppIpRemotePrimaryWinsAddress	jnxPppIpEntry 8	Primary WINS server used by the remote side.
jnxPppIpRemoteSecondaryWinsAddress	jnxPppIpEntry 9	Secondary WINS server used by the remote side.
jnxPppIpNetworkStatusIpcpRenegoTerminates	jnxPppIpEntry 10	Number of times IPCP was terminated because peer exceeded the maximum renegotiation attempts.

jnxPppIpConfigTable

jnxPppIpConfigTable, whose object identifier is **{jnxPppIp 2}**, contains the objects listed in [Table 281 on page 698](#) and provides IPCP configuration information for a particular PPP link.

Each **jnxPpplpConfigEntry**, whose object identifier is **{jnxPpplpConfigTable 1}**, maps to a specific IP parameter for the local PPP entity.

Table 281: jnxPpplpConfigTable

Object	Object ID	Description
jnxPpplpConfigPeerDnsPriority	jnxPpplpConfigEntry 1	<p>Determines whose DNS address prevails in the event of a negotiation conflict. Possible values:</p> <ul style="list-style-type: none"> enable - Peer's DNS address prevails. disable - Local PPP interface DNS address prevails.
jnxPpplpConfigPeerWinsPriority	jnxPpplpConfigEntry 2	<p>Determines whose WINS address prevails in the event of a negotiation conflict. Possible values:</p> <ul style="list-style-type: none"> enable - Peer's WINS address prevails. disable - Local PPP interface WINS address prevails.
jnxPpplpConfigIpcpNetmask	jnxPpplpConfigEntry 3	Enables the negotiation of the IPCP option netmask during IPCP negotiation.
jnxPpplpConfigInitiatelp	jnxPpplpConfigEntry 4	Enables the initiation of negotiation of the IPCP.
jnxPpplpConfigMaxIpcpRenegotiation	jnxPpplpConfigEntry 5	<p>Specifies the maximum number of allowed IPCP renegotiation attempts from peer.</p> <p>NOTE: This attribute is not supported. The attribute returns the default value.</p>
jnxPpplpConfigPromptIpcpDnsOption	jnxPpplpConfigEntry 6	<p>Controls prompting of IPCP DNS option to remote peer.</p> <p>NOTE: This attribute is not supported. The attribute returns the default value.</p>
jnxPpplpConfigIpcpLockout	jnxPpplpConfigEntry 7	<p>Enables IPCP lockout. IPCP negotiation is blocked after a different NCP service is up.</p> <p>NOTE: This attribute is not supported. The attribute returns the default value.</p>

Managed Objects for PPP OSI Network Control Protocol Layer

jnxPppOsiTable and **jnxPppOsiConfigTable**, whose object identifier is **{jnxPPPOObjects 4}**, contains the objects listed in [Table 282 on page 699](#) and [Table 283 on page 701](#), respectively. You can use these managed objects to retrieve information about the status of the NCP state machine as well as the outcome of the option parameter negotiation. The managed

objects are used to manage the PPP Network Control Protocol for OSI protocol operation (OSICP).

For information about **jnxPppOsiTable** and **jnxPppOsiConfigTable**, see the following sections:

- [jnxPppOsiTable on page 699](#)
- [jnxPppOsiConfigTable on page 700](#)

jnxPppOsiTable

jnxPppOsiTable, whose object identifier is **{jnxPppOsi 1}**, contains the objects listed in [Table 282 on page 699](#) and provides status information of the NCP state machine.

Each **jnxPppOsiEntry**, whose object identifier is **{jnxPppOsiTable 1}**, maps to a specific OSI parameter for the local PPP entity.

Table 282: jnxPppOsiTable

Object	Object ID	Description
jnxPppOsiServiceStatus	jnxPppOsiEntry 1	Indication as to whether OSI protocol service is operating over this PPP link.
jnxPppOsiOperStatus	jnxPppOsiEntry 2	Operational status of the OSI network protocol. Possible values: <ul style="list-style-type: none"> • Opened (1) - State machine status is open. • notOpened (2) - State machine status is not open.
jnxPppOsiTerminateReason	jnxPppOsiEntry 3	Reason the OSICP link was terminated. Possible values: <ul style="list-style-type: none"> • none (0) - None. • other (1) - Not specified. • noService (2) - No OSI service was configured on this PPP link. • admin (3) - Administratively disabled. • linkDown (4) - Underlying link is down. • peerTerminated (5) - Peer initiated termination. • peerRenegotiated (6) - Peer initiated renegotiation. • maxRetriesExceeded (7) - Configured maximum number of retries exceeded. • negotiationFailure (8) - Failed to negotiate IPCP option.
jnxPppOsiTerminateNegFailOption	jnxPppOsiEntry 4	PPP OSICP option for which negotiation failed when jnxPppOsiTerminateReason has a value of negotiationFailure (8).

Table 282: jnxPppOsiTable (*continued*)

Object	Object ID	Description
jnxPppOsiLocalAlignNpdu	jnxPppOsiEntry 5	<p>Local alignment of network PDU. Possible values:</p> <ul style="list-style-type: none"> • none (0) - No alignment specified. • oneModulo4 (1) - Alignment on first octet. • twoModulo4 (2) - Alignment on second octet. • threeModulo4 (3) - Alignment on third octet. • fourModulo4 (4) - Alignment on fourth octet. • even (254) - Alignment on even-octet boundary. • odd (255) - Alignment on odd-octet boundary.
jnxPppOsiRemoteAlignNpdu	jnxPppOsiEntry 6	<p>Remote Alignment of network PDU. Possible values:</p> <ul style="list-style-type: none"> • none (0) - No alignment specified. • oneModulo4 (1) - Alignment on first octet. • twoModulo4 (2) - Alignment on second octet. • threeModulo4 (3) - Alignment on third octet. • fourModulo4 (4) - Alignment on fourth octet. • even (254) - Alignment on even-octet boundary. • odd (255) - Alignment on odd-octet boundary.

jnxPppOsiConfigTable

jnxPppOsiConfigTable, whose object identifier is **{jnxPppOsi 2}**, contains the objects listed in [Table 283 on page 701](#), and provides administrative control over the NCP state machine and permits configuration of option parameters that can be used during NCP negotiation.

Each **jnxPppOsiConfigEntry**, whose object identifier is **{jnxPppOsiConfigTable 1}**, maps to a specific option parameter that can be used for NCP negotiation.

Table 283: jnxPppOsiConfigTable

Object	Object ID	Description
jnxPppOsiConfigAdminStatus	jnxPppOsiConfigEntry 1	<p>Status of the OSI network protocol. Possible values:</p> <ul style="list-style-type: none"> • open (1) - Administrative status of the NCP state machine changes to open. • close (2) - Administrative status of the NCP state machine changes to closed.

Managed Objects for PPP Session Attributes

jnxPppSessionTable, whose object identifier is **{jnxPppSession 1}**, contains the objects listed in [Table 284 on page 701](#). You can use the managed objects to retrieve information about the status of each PPP session.

Each **jnxPppSessionEntry**, whose object identifier is **{jnxPppSessionTable 1}**, maps to a specific characteristic of the PPP interface.

Table 284: jnxPppSessionTable

Object	Object ID	Description
jnxPppSessionGrant	jnxPppSessionEntry 1	Indication whether the session was authenticated.
jnxPppSessionterminateReason	jnxPppSessionEntry 2	<p>Reason the session was terminated. Possible values:</p> <ul style="list-style-type: none"> • none (0) • unknown (1) • userRequest (2) • keepaliveFailure (3) • sessionTimeout (4) • inactivityTimeout (5) • adminDisable (6) • lowerLayerDown (7) • noUpperInterface (8) • deny (9) • noHardware (10) • noResources (11) • noInterface (12) • challengeTimeout (13) • requestTimeout (14) • authenticatorTimeout (15) • addressLeaseExpired (16) • adminLogout (17) • tunnelFailed (18)
jnxPppSessionStartTime	jnxPppSessionEntry 3	Value of the sysUpTime when this session last became active.

Table 284: jnxPppSessionTable (*continued*)

Object	Object ID	Description
jnxPppSessionSessionTimeout	jnxPppSessionEntry 8	Maximum duration for the session, after which the session automatically terminates.
jnxPppSessionInactivityTimeout	jnxPppSessionEntry 9	Maximum inactivity duration for the session, after which the session automatically terminates.
jnxPppSessionAccountingInterval	jnxPppSessionEntry 10	Interval that must elapse between generation of accounting records for this session.
jnxPppSessionRemoteIpAddress	jnxPppSessionEntry 11	IP address of the remote PPP entity obtained from the authentication service for IPCP negotiation.
jnxPppSessionRemotePrimaryDnsAddress	jnxPppSessionEntry 12	IP address of the remote primary DNS server obtained from the authentication service for IPCP negotiation.
jnxPppSessionRemoteSecondaryDnsAddress	jnxPppSessionEntry 13	IP address of the remote secondary DNS server obtained from the authentication service for IPCP negotiation.
jnxPppSessionRemotePrimaryWinsAddress	jnxPppSessionEntry 14	IP address of the remote primary WINS server obtained from the authentication service for IPCP negotiation.
jnxPppSessionRemoteSecondaryWinsAddress	jnxPppSessionEntry 15	IP address of the remote primary WINS server obtained from the authentication service for IPCP negotiation.
jnxPppSessionRemoteIpv6AddressIdentifier	jnxPppSessionEntry 16	IPv6 address interface identifier obtained from the authentication service for IPCP negotiation.
jnxPppSessionInhibitIp	jnxPppSessionEntry 17	Indication whether the IP service is inhibited by the authentication service for this session.
jnxPppSessionInhibitIpv6	jnxPppSessionEntry 18	Indication whether the IPv6 service is inhibited by the authentication service for this session.
jnxPppSessionInOctets64	jnxPppSessionEntry 19	Number of octets received since this session last became active.
jnxPppSessionOutOctets64	jnxPppSessionEntry 20	Number of octets sent since this session last became active.
jnxPppSessionInPackets64	jnxPppSessionEntry 21	Number of packets received since this session last became active.
jnxPppSessionOutPackets64	jnxPppSessionEntry 22	Number of packets sent since this session last became active.

Managed Objects for Multilink PPP Layer

The `jnxPppMLPpp` object, whose object identifier is `{jnxPPPObjects 6}`, defines the managed objects listed in [Table 286 on page 704](#), along with the objects listed in `jnxPppMLPppBundleTable`, `jnxPppMLPppLinkConfigTable`, `jnxPppMLPppNetworkConfigTable`, `jnxPppMLPppNextNetworkIfIndex`, and `jnxPppMLPppLinkBindTable`. You can use these managed objects to retrieve information about the MLPPP bundles present in the system.

For information about `jnxPppMLPppBundleTable`, `jnxPppMLPppNextLinkIfIndex`, `jnxPppMLPppLinkConfigTable`, `jnxPppMLPppNetworkConfigTable`, `jnxPppMLPppNextNetworkIfIndex`, and `jnxPppMLPppLinkBindTable`, see the following sections:

- [jnxPppMLPppBundleTable on page 703](#)
- [jnxPppMLPppNextLinkIfIndex on page 703](#)
- [jnxPppMLPppLinkConfigTable on page 704](#)
- [jnxPppMLPppNextNetworkIfIndex on page 706](#)
- [jnxPppMLPppNetworkConfigTable on page 707](#)
- [jnxPppMLPppLinkBindTable on page 707](#)

`jnxPppMLPppBundleTable`

`jnxPppMLPppBundleTable`, whose object identifier is `{jnxPppMLPpp 1}`, contains the objects listed in [Table 285 on page 703](#) and provides information about the characteristics of the MLPPP bundle.

Each `jnxPppMLPppBundleEntry`, whose object identifier is `{jnxPppMLPppBundleTable 1}`, maps to a specific characteristic of the MLPPP bundle in the system.

Table 285: `jnxPppMLPppBundleTable`

Object	Object ID	Description
<code>jnxPppMLPppBundleName</code>	<code>jnxPppMLPppBundleEntry 1</code>	Administrative name of the MLPPP bundle associated with this MLPPP network interfaces.
<code>jnxPppMLPppBundleRowStatus</code>	<code>jnxPppMLPppBundleEntry 2</code>	Row status for this entry. Possible values: <ul style="list-style-type: none"> • <code>createAndGo</code> (4) • <code>destroy</code> (6)
<code>jnxPppMLPppBundleNetworkIfIndex</code>	<code>jnxPppMLPppBundleEntry 3</code>	IfIndex of the MLPPP network interface.

`jnxPppMLPppNextLinkIfIndex`

The `jnxPppMLPppNextLinkIfIndex` object, whose object identifier is `{jnxPppMLPpp 2}`, contains the object listed in [Table 286 on page 704](#). Successive `Get` requests usually return

different values, to avoid collisions among clients seeking to create table entries simultaneously.

Table 286: jnxPppMLPppNextLinkIfIndex

Object	Object ID	Description
jnxPppMLPppNextLinkIfIndex	jnxPppMLPpp 2	Coordinates allocation of ifIndex value for entries in jnxPppMLPppLinkConfigTable.

jnxPppMLPppLinkConfigTable

jnxPppMLPppLinkConfigTable, whose object identifier is {jnxPppMLPpp 3}, contains the objects listed in [Table 287 on page 704](#) and provides information about the MLPPP interfaces present in the system.

Each jnxPppMLPppLinkConfigEntry, whose object identifier is {jnxPppMLPppLinkConfigTable 1}, maps to a specific characteristic of the MLPPP interface present in the system.

Table 287: jnxPppMLPppLinkConfigTable

Object	Object ID	Description
jnxPppMLPppLinkConfigIfIndex	jnxPppMLPppLinkConfigEntry 1	IfIndex of the MLPPP interface.
jnxPppMLPppLinkConfigLowerIfIndex	jnxPppMLPppLinkConfigEntry 2	IfIndex of the interface over which this MLPPP interface is layered. Zero indicates no layering.
jnxPppMLPppLinkConfigKeepalive	jnxPppMLPppLinkConfigEntry 3	Keepalive duration, in seconds. Value of zero disables keepalive.
jnxPppMLPppLinkConfigAuthentication	jnxPppMLPppLinkConfigEntry 4	Type of authentication, to be negotiated with a peer. Possible values: <ul style="list-style-type: none"> • none - No authentication is negotiated. • pap - PAP negotiation only. • chap - CHAP negotiation only. • papChap - PAP negotiation is attempted first; if that fails, CHAP negotiation is attempted. • chapPap - CHAP negotiation is attempted first; if that fails, PAP negotiation is attempted.
jnxPppMLPppLinkConfigMaxAuthenRetries	jnxPppMLPppLinkConfigEntry 5	Number of authentication retries permitted, in addition to a failed initial attempt.

Table 287: jnxPppMLPppLinkConfigTable (*continued*)

Object	Object ID	Description
jnxPppMLPppLinkConfigRowStatus	jnxPppMLPppLinkConfigEntry 6	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo (4) • destroy (6)
jnxPppMLPppLinkConfigAaaProfile	jnxPppMLPppLinkConfigEntry 7	Name of the access profile used for authentication on the MLPPP interface.
jnxPppMLPppLinkConfigChapMinChallengeLength	jnxPppMLPppLinkConfigEntry 8	Minimum value of the CHAP authenticator challenge length value.
jnxPppMLPppLinkConfigChapMaxChallengeLength	jnxPppMLPppLinkConfigEntry 9	Maximum value of the CHAP authenticator challenge length value.
jnxPppMLPppLinkConfigPassiveMode	jnxPppMLPppLinkConfigEntry 10	Status of the MLPPP state machine. Possible values: <ul style="list-style-type: none"> • enabled - Status of the MLPPP state machine moves to passive mode. • disabled - Status of the MLPPP state machine does not move to passive mode.
jnxPppMLPppLinkConfigAuthenticatorLogicalSystem	jnxPppMLPppLinkConfigEntry 11	Name of the logical system used for authentication on the MLPPP interface.
jnxPppMLPppLinkConfigAuthenticatorRoutingInstance	jnxPppMLPppLinkConfigEntry 12	Name of the routing instance used for authentication on the MLPPP interface.
jnxPppMLPppLinkConfigFragmentation	jnxPppMLPppLinkConfigEntry 13	Status of MLPPP fragmentation. Possible values: <ul style="list-style-type: none"> • enabled - Fragmentation is enabled. • disabled - Fragmentation is disabled.
jnxPppMLPppLinkConfigReassembly	jnxPppMLPppLinkConfigEntry 14	Status of MLPPP reassembly. <ul style="list-style-type: none"> • enabled - Reassembly is enabled. • disabled - Reassembly is disabled.
jnxPppMLPppLinkConfigMaxReceiveReconstructedUnit	jnxPppMLPppLinkConfigEntry 15	Maximum receive reconstructed unit (MRRU) that the local MLPPP entity advertises to the remote MLPPP entity.

Table 287: jnxPppMLPppLinkConfigTable (*continued*)

Object	Object ID	Description
jnxPppMLPppLinkConfigFragmentSize	jnxPppMLPppLinkConfigEntry 16	Size of the fragments transmitted by the local MLPpp entity.
jnxPppMLPppLinkConfigHashLinkSelection	jnxPppMLPppLinkConfigEntry 17	<p>Status of MLPpp link selection for non-best effort traffic.</p> <ul style="list-style-type: none"> • enabled - MLPpp hash-based link selection for non-best traffic. • disabled - MLPpp hash-based link selection is not selected.
jnxPppMLPppLinkConfigAuthentication2	jnxPppMLPppLinkConfigEntry 18	<p>Type of authentication. Possible values:</p> <ul style="list-style-type: none"> • none (0) • pap (1) • chap (2) • eap (3)
jnxPppMLPppLinkConfigIgnoreMagicNumberMismatch	jnxPppMLPppLinkConfigEntry 19	<p>Action to be taken when the peer has not negotiated yet or has sent either null or invalid magic number in the LCP packet. Possible values:</p> <ul style="list-style-type: none"> • Ignore the mismatch and retain the connection (1) • Disallow the match and terminate the connection (2)
jnxPppMLPppLinkConfigMultilinkMaxMultiClass	jnxPppMLPppLinkConfigEntry 20	Status of multiclass multilink PPP. Possible values: Enabled and disabled.
jnxPppMLPppLinkConfigMultilinkMaxMultiClasses	jnxPppMLPppLinkConfigEntry 21	Maximum number of MCML classes to be negotiated.

jnxPppMLPppNextNetworkIfIndex

The **jnxPppMLPppNextNetworkIfIndex** object, whose object identifier is {jnxPppMLPpp 4}, contains the object listed in [Table 288 on page 706](#). Successive **Get** requests usually return different values, to avoid collisions among clients seeking to create table entries simultaneously.

Table 288: jnxPppMLPppNextNetworkIfIndex

Object	Object ID	Description
jnxPppMLPppNextNetworkIfIndex	jnxPppMLPpp 4	Coordinates allocation of ifIndex value for entries in jnxPppMLPppNetworkConfigTable .

jnxPppMLPppNetworkConfigTable

jnxPppMLPppNetworkConfigTable, whose object identifier is **{jnxPppMLPpp 5}**, contains the objects listed in [Table 289 on page 707](#) and provides information about the characteristics of the MLPPP network interfaces present in the system.

Each **jnxPppMLPppNetworkConfigEntry**, whose object identifier is **{jnxPppMLPppNetworkConfigTable 1}**, maps to a specific characteristic of the MLPPP network interface in the system.

Table 289: jnxPppMLPppNetworkConfigTable

Object	Object ID	Description
jnxPppMLPppNetworkConfigIfIndex	jnxPppMLPppNetworkConfigEntry 1	IfIndex of the MLPPP network interface.
jnxPppMLPppNetworkConfigLowerIfIndex	jnxPppMLPppNetworkConfigEntry 2	IfIndex of a PPP link interface over which this MLPPP network interface is to be layered.
jnxPppMLPppNetworkBundleName	jnxPppMLPppNetworkConfigEntry 3	Administrative name of the MLPPP bundle.
jnxPppMLPppNetworkRowStatus	jnxPppMLPppNetworkConfigEntry 4	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy

jnxPppMLPppLinkBindTable

jnxPppMLPppLinkBindTable, whose object identifier is **{jnxPppMLPpp 6}**, contains the objects listed in [Table 290 on page 707](#) and provides information about the characteristics of the MLPPP link interface and MLPPP network interfaces bindings.

Each **jnxPppMLPppLinkBindEntry**, whose object identifier is **{jnxPppMLPppLinkBindTable 1}**, maps to a specific characteristic of the MLPPP link interface.

Table 290: jnxPppMLPppLinkBindTable

Object	Object ID	Description
jnxPppMLPppBindNetworkIfIndex	jnxPppMLPppLinkBindEntry 1	IfIndex of the MLPPP network interface.
jnxPppMLPppBindLinkIfIndex	jnxPppMLPppLinkBindEntry 2	IfIndex of the MLPPP link interface bound by the MLPPP network interface.
jnxPppMLPppBindRowStatus	jnxPppMLPppLinkBindEntry 3	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy

Managed Objects for PPP Interface Summary Counts

The PPP interface summary counts, whose object identifier is **{jnxPPPObjects 7}**, contains the managed objects listed in [Table 291 on page 708](#). You can retrieve information about the number of PPP interfaces configured or created, as well as the status of the interfaces using the managed objects listed in [Table 291 on page 708](#).

Each **jnxPppSummary** maps to a specific characteristic of the PPP interface configured on the system.

Table 291: jnxPppSummaryCounts

Object	Object ID	Description
jnxPppSummaryPppInterfaceCount	jnxPppSummary 1	Total number of PPP interfaces configured in the system.
jnxPppSummaryPppIpNCPs	jnxPppSummary 2	Total number of IP NCPs configured in the system.
jnxPppSummaryPppOsiNCPs	jnxPppSummary 3	Total number of OSI NCPs configured in the system.
jnxPppSummaryPppIfAdminUp	jnxPppSummary 4	Total number of PPP interfaces in the system with administrative status of up.
jnxPppSummaryPppIfAdminDown	jnxPppSummary 5	Total number of PPP interfaces in the system with administrative status of down.
jnxPppSummaryPppIfOperUp	jnxPppSummary 6	Total number of PPP interfaces in the system with operational status of up.
jnxPppSummaryPppIfOperDown	jnxPppSummary 7	Total number of PPP interfaces in the system with operational state of down.
jnxPppSummaryPppIfOperDormant	jnxPppSummary 8	Total number of PPP interfaces in the system with operational status of dormant.
jnxPppSummaryPppIfNotPresent	jnxPppSummary 9	Total number of PPP interfaces in the system with operational status of not present.
jnxPppSummaryPppIfLowerLayerDown	jnxPppSummary 10	Total number of PPP interfaces in the system with operational status of lower layer down.
jnxPppSummaryPppIpNcpOpened	jnxPppSummary 11	Total number of PPP IP NCPs in the system with an operational status of open.
jnxPppSummaryPppIpNcpClosed	jnxPppSummary 12	Total number of PPP interfaces in the system with an operational status of closed.
jnxPppSummaryPppOsiNcpOpened	jnxPppSummary 13	Total number of PPP OSI NCPs in the system with an operational status of open.

Table 291: jnxPppSummaryCounts (*continued*)

Object	Object ID	Description
jnxPppSummaryPppOsiNcpClosed	jnxPppSummary 14	Total number of PPP OSI NCPs in the system with an operational status of closed.
jnxPppSummaryPppIfLastChangeTime	jnxPppSummary 15	Value of the sysUpTime since the last creation or deletion of a PPP interface in the system. Zero indicates that the number of PPP interfaces remains unchanged since the last reinitialization.
jnxPppSummaryPppLinkInterfaceCount	jnxPppSummary 16	Total number of PPP link interfaces configured in the system.
jnxPppSummaryPppLinkIfAdminUp	jnxPppSummary 17	Total number of PPP link interfaces in the system with administrative status of up.
jnxPppSummaryPppLinkIfAdminDown	jnxPppSummary 18	Total number of PPP link interfaces in the system with administrative status of down.
jnxPppSummaryPppLinkIfOperUp	jnxPppSummary 19	Total number of PPP link interfaces in the system with an operational status of up.
jnxPppSummaryPppLinkIfOperDown	jnxPppSummary 20	Total number of PPP link interfaces in the system with an operational status of down.
jnxPppSummaryPppLinkIfOperDormant	jnxPppSummary 21	Total number of PPP link interfaces in the system with an operational status of dormant.
jnxPppSummaryPppLinkIfNotPresent	jnxPppSummary 22	Total number of PPP link interfaces in the system with an operational status of not present.
jnxPppSummaryPppLinkIfLowerLayerDown	jnxPppSummary 23	Total number of PPP link interfaces in the system with an operational status of lower layer down.
jnxPppSummaryPppLinkIfLastChangeTime	jnxPppSummary 24	Value of the sysUpTime since the last creation or deletion of a PPP interface in the system. Zero indicates that the number of PPP interfaces remains unchanged since the last reinitialization.
jnxPppSummaryPppNetworkInterfaceCount	jnxPppSummary 25	Total number of PPP network interfaces configured in the system.
jnxPppSummaryPppNetworkIpNCPs	jnxPppSummary 26	Total number of IP NCPs in the system configured on PPP network interfaces.
jnxPppSummaryPppNetworkOsiNCPs	jnxPppSummary 27	Total number of OSI NCPs in the system configured on PPP network interfaces.
jnxPppSummaryPppNetworkIfAdminUp	jnxPppSummary 28	Total number of PPP network interfaces in the system with an administrative status of up.

Table 291: jnxPppSummaryCounts (*continued*)

Object	Object ID	Description
jnxPppSummaryPppNetworkIfAdminDown	jnxPppSummary 29	Total number of PPP network interfaces in the system with an administrative status of down.
jnxPppSummaryPppNetworkIfOperUp	jnxPppSummary 30	Total number of PPP network interfaces in the system with an operational status of up.
jnxPppSummaryPppNetworkIfOperDown	jnxPppSummary 31	Total number of PPP network interfaces in the system with an operational status of down.
jnxPppSummaryPppNetworkIfOperDormant	jnxPppSummary 32	Total number of PPP network interfaces in the system with an operational status of dormant.
jnxPppSummaryPppNetworkIfNotPresent	jnxPppSummary 33	Total number of PPP network interfaces in the system with an operational status of not present.
jnxPppSummaryPppNetworkIfLowerLayerDown	jnxPppSummary 34	Total number of PPP network interfaces in the system with an operational status of lower layer down.
jnxPppSummaryPppNetworkIpNcpOpened	jnxPppSummary 35	Total number of PPP IP NCPs in the system with an operational status of open.
jnxPppSummaryPppNetworkIpNcpClosed	jnxPppSummary 36	Total number of PPP IP NCPs in the system with an operational status of closed.
jnxPppSummaryPppNetworkOsiNcpOpened	jnxPppSummary 37	Total number of PPP OSI NCPs in the system with an operational status of open.
jnxPppSummaryPppNetworkOsiNcpClosed	jnxPppSummary 38	Total number of PPP OSI NCPs in the system with an operational status of closed.
jnxPppSummaryPppNetworkIfLastChangeTime	jnxPppSummary 39	Value of the sysUpTime since the last creation or deletion of a PPP interface in the system. Zero indicates that the number of PPP interfaces remains unchanged since the last reinitialization.
jnxPppSummaryPppIpv6NCPs	jnxPppSummary 40	Total number of IPv6 NCPs configured in the system.
jnxPppSummaryPppIpv6NcpOpened	jnxPppSummary 41	Total number of IPv6 NCPs configured in the system with an operational status of open.
jnxPppSummaryPppIpv6NcpClosed	jnxPppSummary 42	Total number of IPv6 NCPs configured in the system with an operational status of closed.
jnxPppSummaryPppNetworkIpv6NCPs	jnxPppSummary 43	Total number of IPv6 NCPs configured in the system.
jnxPppSummaryPppNetworkIpv6NcpOpened	jnxPppSummary 44	Total number of PPP IPv6 NCPs configured in the system with an operational status of open.

Table 291: jnxPppSummaryCounts (*continued*)

Object	Object ID	Description
jnxPppSummaryPppNetworkIpv6NcpClosed	jnxPppSummary 45	Total number of PPP IPv6 NCPs configured in the system with an operational status of closed.
jnxPppSummaryPppStaticInterfaceCount	jnxPppSummary 46	Total number of static PPP interfaces configured in the system.
jnxPppSummaryPppMplsNCPs	jnxPppSummary 47	Total number of MPLS NCPs configured in the system.
jnxPppSummaryPppIpAdminOpen	jnxPppSummary 48	Total number of IP NCPs in the system with an administrative status of open.
jnxPppSummaryPppIpAdminClose	jnxPppSummary 49	Total number of IP NCPs in the system with an administrative status of closed.
jnxPppSummaryPppIpv6AdminOpen	jnxPppSummary 50	Total number of IPv6 NCPs in the system with an administrative status of open.
jnxPppSummaryPppIpv6AdminClose	jnxPppSummary 51	Total number of IPv6 NCPs in the system with an administrative status of closed.
jnxPppSummaryPppOsiAdminOpen	jnxPppSummary 52	Total number of OSI NCPs in the system with an administrative status of open.
jnxPppSummaryPppOsiAdminClose	jnxPppSummary 53	Total number of OSI NCPs in the system with an administrative status of closed.
jnxPppSummaryPppMplsAdminOpen	jnxPppSummary 54	Total number of MPLS NCPs in the system with an administrative status of open.
jnxPppSummaryPppMplsAdminClose	jnxPppSummary 55	Total number of MPLS NCPs in the system with an administrative status of closed.
jnxPppSummaryPppIpNcpNotPresent	jnxPppSummary 56	Total number of PPP IP NCPs in the system with an operational status of not present.
jnxPppSummaryPppIpNcpNoResources	jnxPppSummary 57	Total number of PPP IP NCPs in the system with an operational status of no resources.
jnxPppSummaryPppIpv6NcpNotPresent	jnxPppSummary 58	Total number of PPP IPv6 NCPs in the system with an operational status of not present.
jnxPppSummaryPppIpv6NcpNoResources	jnxPppSummary 59	Total number of PPP IPv6 NCPs in the system with an operational status of no resources.
jnxPppSummaryPppOsiNcpNotPresent	jnxPppSummary 60	Total number of PPP OSI NCPs in the system with an operational status of not present.
jnxPppSummaryPppOsiNcpNoResources	jnxPppSummary 61	Total number of PPP OSI NCPs in the system with an operational status of no resources.

Table 291: jnxPppSummaryCounts (*continued*)

Object	Object ID	Description
jnxPppSummaryPppMplsNcpOpened	jnxPppSummary 62	Total number of PPP MPLS NCPs in the system with an operational status of open.
jnxPppSummaryPppMplsNcpClosed	jnxPppSummary 63	Total number of PPP MPLS NCPs in the system with an operational status of closed.
jnxPppSummaryPppMplsNcpNotPresent	jnxPppSummary 64	Total number of PPP MPLS NCPs in the system with an operational status of not present.
jnxPppSummaryPppMplsNcpNoResources	jnxPppSummary 65	Total number of PPP MPLS NCPs in the system with an operational status of no resources.
jnxPppSummaryPppLinkStaticInterfaceCount	jnxPppSummary 66	Total number of static PPP link interfaces in the system.
jnxPppSummaryPppNetworkStaticInterfaceCount	jnxPppSummary 67	Total number of static PPP network interfaces in the system.
jnxPppSummaryPppNetworkMplsNCPs	jnxPppSummary 68	Total number of MPLS NCPs configured on PPP network interfaces in the system.
jnxPppSummaryPppNetworkIpAdminOpen	jnxPppSummary 69	Total number of IP NCPs configured on PPP network interfaces with an administrative status of open.
jnxPppSummaryPppNetworkIpAdminClose	jnxPppSummary 70	Total number of IP NCPs configured on PPP network interfaces with an administrative status of closed.
jnxPppSummaryPppNetworkIpv6AdminOpen	jnxPppSummary 71	Total number of IPv6 NCPs configured on PPP network interfaces with an administrative status of open.
jnxPppSummaryPppNetworkIpv6AdminClose	jnxPppSummary 72	Total number of IPv6 NCPs configured on PPP network interfaces with an administrative status of closed.
jnxPppSummaryPppNetworkOsiAdminOpen	jnxPppSummary 73	Total number of OSI NCPs configured on PPP network interfaces with an administrative status of open.
jnxPppSummaryPppNetworkOsiAdminClose	jnxPppSummary 74	Total number of OSI NCPs configured on PPP network interfaces with an administrative status of closed.
jnxPppSummaryPppNetworkMplsAdminOpen	jnxPppSummary 75	Total number of MPLS NCPs configured on PPP network interfaces with an administrative status of open.

Table 291: jnxPppSummaryCounts (*continued*)

Object	Object ID	Description
jnxPppSummaryPppNetworkMplsAdminClose	jnxPppSummary 76	Total number of MPLS NCPs configured on PPP network interfaces with an administrative status of closed.
jnxPppSummaryPppNetworkIpNcpNotPresent	jnxPppSummary 77	Total number of IP NCPs configured on PPP network interfaces with an operational status of not present.
jnxPppSummaryPppNetworkIpNcpNoResources	jnxPppSummary 78	Total number of IP NCPs configured on PPP network interfaces with an operational status of no resources.
jnxPppSummaryPppNetworkIpv6NcpNotPresent	jnxPppSummary 79	Total number of IPv6 NCPs configured on PPP network interfaces with an operational status of not present.
jnxPppSummaryPppNetworkIpv6NcpNoResources	jnxPppSummary 80	Total number of IPv6 NCPs configured on PPP network interfaces with an operational status of no resources.
jnxPppSummaryPppNetworkOsiNcpNotPresent	jnxPppSummary 81	Total number of OSI NCPs configured on PPP network interfaces with an operational status of not present.
jnxPppSummaryPppNetworkOsiNcpNoresources	jnxPppSummary 82	Total number of OSI NCPs configured on PPP network interfaces with an operational status of no resources.
jnxPppSummaryPppNetworkMplsNcpOpened	jnxPppSummary 83	Total number of MPLS NCPs configured on PPP network interfaces with an operational status of open.
jnxPppSummaryPppNetworkMplsNcpClosed	jnxPppSummary 84	Total number of MPLS NCPs configured on PPP network interfaces with an operational status of closed.
jnxPppSummaryPppNetworkMplsNcpNotPresent	jnxPppSummary 85	Total number of MPLS NCPs configured on PPP network interfaces with an operational status of not present.
jnxPppSummaryPppNetworkMplsNcpNoResources	jnxPppSummary 86	Total number of MPLS NCPs configured on PPP network interfaces with an operational status of no resources.

Managed Objects for PPP IPv6 Network Control Protocol Layer

jnxPppIpv6Table and jnxPppIpv6ConfigTable, whose object identifier is {jnxPPPObjects 8}, contains the objects listed in [Table 292 on page 714](#) and [Table 293 on page 715](#), respectively. You can use these managed objects to retrieve information about the IPv6CP status information for a specific PPP link.

For information about **jnxPpplpv6Table** and **jnxPpplpv6ConfigTable**, see the following sections:

- [jnxPpplpv6Table on page 714](#)
- [jnxPpplpv6ConfigTable on page 715](#)

jnxPpplpv6Table

jnxPpplpv6Table, whose object identifier is **{jnxPpplpv6 1}**, contains the objects listed in [Table 292 on page 714](#) and provides IPv6CP status information for a particular PPP link.

Each **jnxPpplpv6Entry**, whose object identifier is **{jnxPpplpv6Table 1}**, maps to a specific IPv6 parameter for the local PPP entity.

Table 292: jnxPpplpv6Table

Object	Object ID	Description
jnxPpplpv6ServiceStatus	jnxPpplpv6Entry 1	Indication as to whether IPv6 protocol service is operating over this PPP link.
jnxPpplpv6OperStatus	jnxPpplpv6Entry 2	Operational status of the IPv6 network protocol.
jnxPpplpv6TerminateReason	jnxPpplpv6Entry 3	Reason the IPv6CP link was terminated. Possible values: <ul style="list-style-type: none"> • none (0) - None. • other (1) - Not specified. • noService (2) - No IPv6 service was configured on this PPP link. • admin (3) - Administratively disabled. • linkDown (4) - Underlying link is down. • peerTerminated (5) - Peer initiated termination. • peerRenegotiated (6) - Peer initiated renegotiation. • maxRetriesExceeded (7) - Configured maximum number of retries exceeded. • negotiationFailure (8) - Failed to negotiate IPv6CP option.
jnxPpplpv6TerminateNegfailOption	jnxPpplpv6Entry 4	PPP IPv6CP option for which the negotiation failed when jnxPpplpv6TerminateReason has a value of negotiationFailure(8).
jnxPpplpv6LocalIpv6AddressIdentifier	jnxPpplpv6Entry 5	IPv6 address interface identifier used by the local side.
jnxPpplpv6RemoteIpv6AddressIdentifier	jnxPpplpv6Entry 6	IPv6 address interface identifier used by the remote side.

Table 292: jnxPppIpv6Table (*continued*)

Object	Object ID	Description
jnxPppIpv6NetworkStatusIpv6cpRenegoTerminates	jnxPppIpv6Entry 7	Number of times IPv6CP was terminated because peer exceeded the maximum renegotiation attempts.

jnxPppIpv6ConfigTable

jnxPppIpv6ConfigTable, whose object identifier is **{jnxPppIpv6 2}**, contains the objects listed in [Table 293 on page 715](#) and provides IPv6CP configuration information for a particular PPP link.

Each **jnxPppIpv6ConfigEntry**, whose object identifier is **{jnxPppIpv6ConfigTable 1}**, maps to a specific IPv6 parameter for the local PPP entity.

Table 293: jnxPppIpv6ConfigTable

Object	Object ID	Description
jnxPppIpv6ConfigAdminStatus	jnxPppIpv6ConfigEntry 1	Desired status of the IPv6 network protocol. Possible values: <ul style="list-style-type: none"> open (1) - Administrative status changes to open. close (2) - Administrative status changes to closed.
jnxPppIpv6ConfigInitiateIpv6	jnxPppIpv6ConfigEntry 2	Initiation of negotiation of the IPv6CP. Possible values: Enabled, Disabled.
jnxPppIpv6ConfigMaxIpv6cpRenegotiation	jnxPppIpv6ConfigEntry 3	Maximum number of permitted IPv6CP renegotiation attempts from peer.

Managed Objects for PPP Global Attributes

[Table 294 on page 715](#) contains information about the non-interface based object used in the enterprise-specific PPP MIB.

Table 294: jnxPppGlobal

Object	Object ID	Description
jnxPppPeerIpAddressOptional	jnxPppGlobalConfig 1	Enables you to ignore conflicts between the IP address requested by the PPP client and the address returned by the RADIUS server during IPNCP negotiation.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)

- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 60

PPPoE MIB

- Interpreting the Enterprise-Specific PPPoE MIB on page 717

Interpreting the Enterprise-Specific PPPoE MIB

- PPPoE MIB on page 717
- PPPoE MIB Textual Conventions on page 718
- Managed Objects for PPPoE Underlying Interfaces Layer Functions on page 718
- Managed Objects for PPPoE Interfaces Layer Functions on page 722
- Managed Objects for PPPoE Summary Counts on page 724
- Managed Objects for PPPoE ServiceName Tables on page 725

PPPoE MIB

The Juniper Networks enterprise-specific PPPoE MIB, whose object ID is `{jnxPppoeMibRoot 1}`, extends SNMP support for Point-to-Point Protocol over Ethernet (PPPoE) modules in Junos OS.

The PPPoE MIB stores PPPoE-related information for both PPPoE interfaces and PPPoE underlying interfaces, such as authentication type, status, interface characteristics, and statistics. You can view this information by using SNMP **get** and **get-next** requests.



NOTE: The PPPoE MIB is supported on the Common Edge PPPoE process, `jpppoed`. SNMP support for PPPoE over Ethernet interfaces is active only if the `jpppoed` process is enabled.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pppoe.txt.

For information about enterprise-specific PPPoE MIB objects, see the following topics:

- PPPoE MIB Textual Conventions on page 718
- Managed Objects for PPPoE Underlying Interfaces Layer Functions on page 718
- Managed Objects for PPPoE Interfaces Layer Functions on page 722

- [Managed Objects for PPPoE Summary Counts on page 724](#)
- [Managed Objects for PPPoE ServiceName Tables on page 725](#)

PPPoE MIB Textual Conventions

[Table 295 on page 718](#) contains information about the textual conventions used in the enterprise-specific PPPoE MIB.

Table 295: PPPoE MIB Textual Conventions

Object	Description	Syntax
<code>jnxPPPoEServiceNameAction</code>	Set of Service-Name action types.	This object uses the following integer values: <ul style="list-style-type: none"> • 0—Drop (no PADO packet sent) • 1—Terminate (PADO packet sent)

Managed Objects for PPPoE Underlying Interfaces Layer Functions

You can use the managed objects listed in [Table 296 on page 718](#) along with the objects listed in `jnxPPPoElfTable`, `jnxPPPoElfStatsTable`, and `jnxPPPoElfLockoutTable` to retrieve information about the PPPoE underlying interfaces.

The `jnxPPPoElfLayer` object, whose object identifier is `{jnxPPPoEObjects 1}`, contains the object listed in [Table 296 on page 718](#). Successive **Get** requests usually return different values to avoid collisions among clients seeking to create table entries simultaneously.

Table 296: jnxPPPoENextIfIndex

Object	Object ID	Description
<code>jnxPPPoENextIfIndex</code>	<code>jnxPPPoElfLayer 1</code>	Coordinates allocation of ifindex value for entries in <code>jnxPPPoElfTable</code> .

For information about `jnxPPPoElfTable`, `jnxPPPoElfStatsTable`, and `jnxPPPoElfLockoutTable`, see the following sections:

- [jnxPPPoElfTable on page 718](#)
- [jnxPPPoElfStatsTable on page 720](#)
- [jnxPPPoElfLockoutTable on page 721](#)

jnxPPPoElfTable

`jnxPPPoElfTable`, whose object identifier is `{jnxPPPoElfLayer 2}`, contains `jnxPPPoElfEntry`, which maps to the parameters for a specific PPPoE underlying interface.

Each `jnxPPPoElfEntry`, whose object identifier is `{jnxPPPoElfTable 1}`, contains the objects listed in [Table 297 on page 719](#).

Table 297: jnxPPPoElfTable

Object	Object ID	Description
jnxPPPoElfIfIndex	jnxPPPoElfEntry 1	IfIndex of the PPPoE underlying interface.
jnxPPPoElfMaxNumSessions	jnxPPPoElfEntry 2	Maximum number of sessions allowed on the PPPoE underlying interface. Zero indicates that the maximum number of sessions allowed is unlimited.
jnxPPPoElfRowStatus	jnxPPPoElfEntry 3	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy
jnxPPPoElfLowerIfIndex	jnxPPPoElfEntry 4	IfIndex of an interface over which this PPPoE underlying interface is to be layered. Zero indicates that there is no layering.
jnxPPPoElfAcName	jnxPPPoElfEntry 5	Name of the AC-NAME tag sent in any PADO packet on this interface.
jnxPPPoElfDupProtect	jnxPPPoElfEntry 6	State of PPPoE duplicate protection. Possible values: <ul style="list-style-type: none"> • On - Allows duplicate MAC addresses. • Off - Does not allow duplicate MAC addresses.
jnxPPPoElfPADIFlag	jnxPPPoElfEntry 7	Action taken when the interface receives a PADI packet. The response can be a PADO packet.
jnxPPPoElfAutoconfig	jnxPPPoElfEntry 8	Whether the PPPoE interface is created dynamically or statically. Possible values: <ul style="list-style-type: none"> • enable (1) - dynamic • disable (2) - static
jnxPPPoElfServiceNameTable	jnxPPPoElfEntry 9	Name of the ServiceNameTable associated with this interface for PADI processing.
jnxPPPoElfPadrRemoteCircuitIdcapture	jnxPPPoElfEntry 10	Whether the remote-circuit-id string is captured and subsequently used as the NAS-Port-Id RADIUS attribute when it arrives as a tag in the PADR packet.

Table 297: jnxPPPoElfTable (continued)

Object	Object ID	Description
jnxPPPoElfMtu	jnxPPPoElfEntry 11	Initial maximum transmit unit (MTU) that the PPPoE underlying interface entity advertises to the remote entity. Possible values: <ul style="list-style-type: none"> 1 - Local PPPoE entity uses the MTU value determined by its underlying media interface. 2 - Local PPPoE entity uses a value determined by the PPPoE Max-Mtu-Tag transmitted from the client in the PADR packet. If no Max-Mtu-tag is received, the value defaults to a maximum of 1494.
jnxPPPoElfLockoutMin	jnxPPPoElfEntry 12	Minimum number of seconds used to specify the duration of the lockout of the client from recognition for the specified interface.
jnxPPPoElfLockoutMax	jnxPPPoElfEntry 13	Maximum number of seconds used to specify the duration of the lockout of the client from recognition for the specified interface.
jnxPPPoElfDynamicProfile	jnxPPPoElfEntry 14	Dynamic profile attachment to this interface.

jnxPPPoElfStatsTable

jnxPPPoElfStatsTable, whose object identifier is {jnxPPPoElfLayer 3}, contains jnxPPPoElfStatsEntry, which displays the statistics for the PPPoE underlying interface.

Each jnxPPPoElfStatsEntry, whose object identifier is {jnxPPPoElfStatsTable 1}, contains the objects listed in Table 298 on page 720.

Table 298: jnxPPPoElfStatsTable

Object	Object ID	Description
jnxPPPoElfStatsRxPADI	jnxPPPoElfStatsEntry 1	Number of PADI packets received.
jnxPPPoElfStatsTxPADO	jnxPPPoElfStatsEntry 2	Number of PADO packets transmitted.
jnxPPPoElfStatsRxPADR	jnxPPPoElfStatsEntry 3	Number of PADR packets received.
jnxPPPoElfStatsTxPADS	jnxPPPoElfStatsEntry 4	Number of PADS packets transmitted.
jnxPPPoElfStatsRxPADT	jnxPPPoElfStatsEntry 5	Number of PADT packets received.
jnxPPPoElfStatsTxPADT	jnxPPPoElfStatsEntry 6	Number of PADT packets transmitted.

Table 298: jnxPPPoElfStatsTable (continued)

Object	Object ID	Description
jnxPPPoElfStatsRxInvVersion	jnxPPPoElfStatsEntry 7	Number of packets received with invalid version number.
jnxPPPoElfStatsRxInvCode	jnxPPPoElfStatsEntry 8	Number of packets received with invalid code.
jnxPPPoElfStatsRxInvTags	jnxPPPoElfStatsEntry 9	Number of packets received with invalid tags.
jnxPPPoElfStatsRxInvSession	jnxPPPoElfStatsEntry 10	Number of packets received with invalid session identifiers. NOTE: This object is now obsolete; PADI and PADR packets have separate counters.
jnxPPPoElfStatsRxInvTypes	jnxPPPoElfStatsEntry 11	Number of packets received with invalid types.
jnxPPPoElfStatsRxInvPackets	jnxPPPoElfStatsEntry 12	Number of invalid packets received.
jnxPPPoElfStatsRxInsufficientResources	jnxPPPoElfStatsEntry 13	Number of session requests that were not honored because of insufficient resources.
jnxPPPoElfStatsTxPADM	jnxPPPoElfStatsEntry 14	Number of PADM packets transmitted.
jnxPPPoElfStatsTxPADN	jnxPPPoElfStatsEntry 15	Number of PADN packets transmitted.
jnxPPPoElfStatsRxInvTagLength	jnxPPPoElfStatsEntry 16	Number of packets received with invalid tag lengths.
jnxPPPoElfStatsRxInvLength	jnxPPPoElfStatsEntry 17	Number of packets received with invalid length.
jnxPPPoElfStatsRxInvPadiSession	jnxPPPoElfStatsEntry 18	Number of PADI packets received with invalid session identifiers.
jnxPPPoElfStatsRxInvPadRSession	jnxPPPoElfStatsEntry 19	Number of PADR packets received with invalid session identifiers.

jnxPPPoElfLockoutTable

jnxPPPoElfLockoutTable, whose object identifier is {jnxPPPoElfLayer 4}, enables you to lock out the client in the event of an error during the creation of the PPPoE underlying interface. To disable the ability to lock out the client from recognition, the value of **jnxPPPoElfLockoutMax** and **jnxPPPoElfLockoutMin** objects must be set to 0.



NOTE: None of the objects listed in [Table 299 on page 722](#) are supported. These attributes return the default value.

Each `jnxPPPoElfLockoutEntry`, whose object identifier is `{jnxPPPoElfLockoutTable 1}`, contains the objects listed in [Table 299 on page 722](#).

Table 299: jnxPPPoElfLockoutTable

Object	Object ID	Description
<code>jnxPPPoElfLockoutClientAddress</code>	<code>jnxPPPoElfLockoutEntry 1</code>	Source MAC address of the client.
<code>jnxPPPoElfLockoutTime</code>	<code>jnxPPPoElfLockoutEntry 2</code>	Duration, in seconds, used to lock out the specified encapsulation type from recognition for the specified interface. Zero indicates that no lockout occurred for the encapsulation type for the specified interface.
<code>jnxPPPoElfLockoutElapsedTime</code>	<code>jnxPPPoElfLockoutEntry 3</code>	Duration, in seconds, of the elapsed time since the specified encapsulation type was locked out from recognition for the specified interface. Zero indicates that no lockout occurred for the encapsulation type for the specified interface.
<code>jnxPPPoElfLockoutNextTime</code>	<code>jnxPPPoElfLockoutEntry 4</code>	Duration, in seconds, to lock out the specified encapsulation type from recognition for the specified interface for the next event that results in a lockout condition. When <code>jnxPPPoElfEnable</code> is enabled, a value of zero indicates that lockout is prevented from occurring for the encapsulation type for the specified interface.

Managed Objects for PPPoE Interfaces Layer Functions

The `jnxPPPoESubIfLayer` object, whose object identifier is `{jnxPPPoEObjects 2}`, contains the object listed in [Table 300 on page 723](#), along with the objects listed in `jnxPPPoESubIfTable` and `jnxPPPoESubIfQueueStatsTable`. You can use these managed objects to retrieve information about the PPPoE Interfaces layer.

For information about `jnxPPPoESubIfNextIndex`, `jnxPPPoESubIfTable`, and `jnxPPPoESubIfQueueStatsTable`, see the following sections:

- [jnxPPPoESubIfNextIndex on page 723](#)
- [jnxPPPoESubIfTable on page 723](#)
- [jnxPPPoESubIfQueueStatsTable on page 724](#)

jnxPPPoESubIfNextIndex

The `jnxPPPoESubIfNextIndex` object, whose object identifier is `{jnxPPPoESubIfLayer 1}` contains the objects listed in [Table 300 on page 723](#). Successive `Get` requests usually return different values to avoid collisions among clients seeking to create table entries simultaneously.

Table 300: jnxPPPoESubIfNextIfIndex

Object	Object ID	Description
<code>jnxPPPoESubIfNextIfIndex</code>	<code>jnxPPPoESubIfLayer 1</code>	Coordinates allocation of <code>ifIndex</code> value for entries in <code>jnxPPPoESubIfTable</code> .

jnxPPPoESubIfTable

`jnxPPPoESubIfTable`, whose object identifier is `{jnxPPPoESubIfLayer 2}`, contains `jnxPPPoESubIfEntry`, which maps to the parameters of the specific PPPoE interface.

Each `jnxPPPoESubIfEntry`, whose object identifier is `{jnxPPPoESubIfTable 1}`, contains the objects listed in [Table 301 on page 723](#).

Table 301: jnxPPPoESubIfTable

Object	Object ID	Description
<code>jnxPPPoESubIfIndex</code>	<code>jnxPPPoESubIfEntry 1</code>	<code>ifIndex</code> of the PPPoE interface.
<code>jnxPPPoESubIfRowStatus</code>	<code>jnxPPPoESubIfEntry 2</code>	Row status for this entry. Possible values: <ul style="list-style-type: none"> <code>createAndGo</code> <code>destroy</code>
<code>jnxPPPoESubIfLowerIfIndex</code>	<code>jnxPPPoESubIfEntry 3</code>	<code>ifIndex</code> of a PPPoE underlying interface over which this PPPoE interface is to be layered. Zero indicates that there is no layering.
<code>jnxPPPoESubIfId</code>	<code>jnxPPPoESubIfEntry 4</code>	Identifier for the PPPoE interface.
<code>jnxPPPoESubIfSessionId</code>	<code>jnxPPPoESubIfEntry 5</code>	Session identifier of the PPPoE interface.
<code>jnxPPPoESubIfMotm</code>	<code>jnxPPPoESubIfEntry 6</code>	Message to send via a PADM packet on the interface when the interface transitions to the <code>IfOperStatusUp</code> state.
<code>jnxPPPoESubIfUrl</code>	<code>jnxPPPoESubIfEntry 7</code>	URL to be sent via a PADM packet on the interface when the interface transitions to the <code>IfOperStatusUp</code> state.

jnxPPPoESubIfQueueStatsTable

jnxPPPoESubIfQueueStatsTable, whose object identifier is {jnxPPPoESubIfLayer 3}, contains jnxPPPoESubIfPerQueueStatsEntry that displays the statistics for the PPPoE interface.

Each jnxPPPoESubIfPerQueueStatsEntry, whose object identifier is {jnxPPPoESubIfQueueStatsTable 1}, contains the objects listed in [Table 302 on page 724](#).

Table 302: jnxPPPoESubIfQueueStatsTable

Object	Object ID	Description
jnxPPPoESubIfQueueIndex	jnxPPPoESubIfPerQueueStatsEntry 1	Queue index of the queue configured on the PPPoE interface. Range of values is 0 through 7.
jnxPPPoESubIfQueueStatsPacketSent	jnxPPPoESubIfPerQueueStatsEntry 2	Number of packets sent per PPPoE session and per queue.
jnxPPPoESubIfQueueStatsBytesSent	jnxPPPoESubIfPerQueueStatsEntry 3	Number of bytes sent per PPPoE session and per queue.
jnxPPPoESubIfQueueStatsPacketDropped	jnxPPPoESubIfPerQueueStatsEntry 4	Number of packets dropped per PPPoE session and per queue.
jnxPPPoESubIfQueueStatsBytesDropped	jnxPPPoESubIfPerQueueStatsEntry 5	Number of bytes dropped per PPPoE session and per queue.
jnxPPPoESubIfQueueStatsActualBitRate	jnxPPPoESubIfPerQueueStatsEntry 6	Actual bit rate for per PPPoE session and per queue.
jnxPPPoESubIfQueueStatsActualDroppedBitRate	jnxPPPoESubIfPerQueueStatsEntry 7	Actual dropped bit rate per PPPoE session and per queue.

Managed Objects for PPPoE Summary Counts

The managed objects listed in [Table 303 on page 724](#) provide information about the number of PPPoE interfaces, PPPoE underlying interfaces configured or created, as well as the status of the interfaces, and underlying interfaces.

Each jnxPPPoESummary, whose object identifier is {jnxPPPoEObjects 3}, contains the objects listed in [Table 303 on page 724](#).

Table 303: jnxPPPoESummaryCounts

Object	Object ID	Description
jnxPPPoEMajorInterfaceCount	jnxPPPoESummary 1	Total number of PPPoE underlying interfaces configured and created in the system.
jnxPPPoESummaryMajorIfAdminUp	jnxPPPoESummary 2	Total number of PPPoE underlying interfaces in the system with administrative status configured as up.

Table 303: jnxPPPoESummaryCounts (*continued*)

Object	Object ID	Description
jnxPPPoESummaryMajorIfAdminDown	jnxPPPoESummary 3	Total number of PPPoE underlying interfaces in the system with administrative status configured as down.
jnxPPPoESummaryMajorIfOperUp	jnxPPPoESummary 4	Total number of PPPoE underlying interfaces in the system whose operating status is up.
jnxPPPoESummaryMajorIfOperDown	jnxPPPoESummary 5	Total number of PPPoE underlying interfaces in the system whose operational status is down.
jnxPPPoESummaryMajorIfLowerLayerDown	jnxPPPoESummary 6	Total number of PPPoE underlying interfaces in the system whose operational status is lowerLayerDown.
jnxPPPoESummaryMajorIfNotPresent	jnxPPPoESummary 7	Total number of PPPoE underlying interfaces whose operational state is notPresent.
jnxPPPoESummarySubInterfaceCount	jnxPPPoESummary 8	Total number of PPPoE interfaces configured in the system.
jnxPPPoESummarySubIfAdminUp	jnxPPPoESummary 9	Total number of PPPoE interfaces in the system whose administrative status is up.
jnxPPPoESummarySubIfAdminDown	jnxPPPoESummary 10	Total number of PPPoE interfaces in the system whose administrative status is down.
jnxPPPoESummarySubIfOperUp	jnxPPPoESummary 11	Total number of PPPoE interfaces in the system with an operational status of up.
jnxPPPoESummarySubIfOperDown	jnxPPPoESummary 12	Total number of PPPoE interfaces in the system whose operational status is down.
jnxPPPoESummarySubIfLowerLayerDown	jnxPPPoESummary 13	Total number of PPPoE interfaces in the system whose operational status is lowerLayerDown.
jnxPPPoESummarySubIfNotPresent	jnxPPPoESummary 14	Total number of PPPoE interfaces in the system whose operational status is notPresent.

Managed Objects for PPPoE ServiceName Tables

You can use the managed objects listed in `jnxPPPoEServiceNameTableTable`, `jnxPPPoEServiceNameTable`, and `jnxPPPoEServiceNameAciAriTable` to retrieve information about the PPPoE service name tables.

For information about `jnxPPPoEServiceNameTableTable`, `jnxPPPoEServiceNameTable`, and `jnxPPPoEServiceNameAciAriTable`, see the following sections:

- [jnxPPPoEServiceNameTableTable on page 726](#)
- [jnxPPPoEServiceNameTable on page 726](#)
- [jnxPPPoEServiceNameAciAriTable on page 727](#)

jnxPPPoEServiceNameTableTable

jnxPPPoEServiceNameTableTable, whose object identifier is **{jnxPPPoEServices 1}**, contains **jnxPPPoEServiceNameTableEntry**, which maps to the objects in the service name table. The Empty service and Any service are automatically configured for each service name table created.

Each **jnxPPPoEServiceNameTableEntry**, whose object identifier is **{jnxPPPoEServiceNameTableTable 1}**, contains the objects listed in [Table 304 on page 726](#).

Table 304: jnxPPPoEServiceNameTableTable

Object	Object ID	Description
jnxPPPoEServiceNameTableName	jnxPPPoEServiceNameTableEntry 1	Name of the service name table.
jnxPPPoEServiceNameTableRowStatus	jnxPPPoEServiceNameTableEntry 2	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy

jnxPPPoEServiceNameTable

jnxPPPoEServiceNameTable, whose object identifier is **{jnxPPPoEServices 2}**, contains **jnxPPPoEServiceNameEntry**, which maps to the parameters of the service name table.

Each **jnxPPPoEServiceNameEntry**, whose object identifier is **{jnxPPPoEServiceNameTable 1}**, contains the objects listed in [Table 305 on page 726](#).

Table 305: jnxPPPoEServiceNameTable

Object	Object ID	Description
jnxPPPoEServiceName	jnxPPPoEServiceNameEntry 1	Service name tag value.
jnxPPPoEServiceNameAction	jnxPPPoEServiceNameEntry 2	Identifies the behavior when the service name tag is received in a PADI frame.
jnxPPPoEServiceNameDynamicProfile	jnxPPPoEServiceNameEntry 3	Name of the dynamic profile associated with the service name.
jnxPPPoEServiceNameRoutingInstance	jnxPPPoEServiceNameEntry 4	Routing instance associated with the service name.
jnxPPPoEServiceNameMaxSessions	jnxPPPoEServiceNameEntry 5	Maximum number of active PPPoE sessions that can be established with the specified service entry.
jnxPPPoEServiceNameRowStatus	jnxPPPoEServiceNameEntry 6	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy

jnxPPPoEServiceNameAciAriTable

jnxPPPoEServiceNameAciAriTable, whose object identifier is **{jnxPPPoEServices 3}**, contains **jnxPPPoEServiceNameAciAriEntry**, which maps to the parameters in the PPPoE Service Name Agent Circuit Identifier (ACI)/ Agent Remote Identifier (ARI) tables.

Each **jnxPPPoEServiceNameAciAriEntry**, whose object identifier is **{jnxPPPoEServiceNameAriAriTable 1}**, contains the objects listed in [Table 306 on page 727](#).

Table 306: jnxPPPoEServiceNameAciAriTable

Object	Object ID	Description
jnxPPPoEServiceNameAgentCircuitId	jnxPPPoEServiceNameAciAriEntry 1	Agent Circuit Identifier tag values that the PPPoE client sends in the PADI or PADR control packet.
jnxPPPoEServiceNameAgentRemotId	jnxPPPoEServiceNameAciAriEntry 2	Agent Circuit Identifier tag values that the PPPoE client sends in the PADI or PADR control packet.
jnxPPPoEServiceNameAciAriAction	jnxPPPoEServiceNameAciAriEntry 3	Identification of the behavior when the service name with ACI/ARI pair is received in a PADI frame.
jnxPPPoEServiceNameAciAriDynamicProfile	jnxPPPoEServiceNameAciAriEntry 4	Dynamic profile associated with a service name and ACI/ARI pair.
jnxPPPoEServiceNameAciAriRoutingInstance	jnxPPPoEServiceNameAciAriEntry 5	Routing instance associated with a service name and ACI/ARI pair.
jnxPPPoEServiceNameAciAriStaticInterface	jnxPPPoEServiceNameAciAriEntry 6	Static interface associated with matching ACI/ARI pair. A static interface can be configured only for an ACI/ARI pair.
jnxPPPoEServiceNameAciAriRowStatus	jnxPPPoEServiceNameAciAriEntry 7	Row status for this entry. Possible values: <ul style="list-style-type: none"> • createAndGo • destroy

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 61

Pseudowire ATM MIB

- [Interpreting the Enterprise-Specific Pseudowire ATM MIB on page 729](#)

Interpreting the Enterprise-Specific Pseudowire ATM MIB

- [Pseudowire ATM MIB on page 729](#)
- [jnxpwAtmCfgTable on page 730](#)
- [jnxpwAtmOutboundNto1Table on page 731](#)
- [jnxpwAtmInboundNto1Table on page 733](#)
- [jnxpwAtmPerfCurrentTable on page 735](#)
- [jnxpwAtmPerfIntervalTable on page 736](#)
- [jnxpwAtmPerf1DayIntervalTable on page 738](#)

Pseudowire ATM MIB

The enterprise-specific Pseudowire ATM MIB is the Juniper Networks implementation of RFC 5605, *Managed Objects for ATM over Packet Switched Networks (PSNs)*.

The Juniper Networks enterprise-specific Pseudowire ATM MIB defines objects used for managing ATM pseudowires in Juniper Networks products.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-pwatm.txt.

Starting with Junos OS Release 12.3, the Juniper Networks enterprise-specific Pseudowire ATM MIB supports the Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP (MIC-3D-4COC3-1COC12-CE).

For information about the Pseudowire ATM MIB, see the following topics:

- [jnxpwAtmCfgTable on page 730](#)
- [jnxpwAtmOutboundNto1Table on page 731](#)
- [jnxpwAtmInboundNto1Table on page 733](#)
- [jnxpwAtmPerfCurrentTable on page 735](#)

- [jnxpwAtmPerfIntervalTable](#) on page 736
- [jnxpwAtmPerfDayIntervalTable](#) on page 738

jnxpwAtmCfgTable

The jnxpwAtmCfgTable, whose object identifier is {jnxpwAtmObjects 1}, specifies generic information for an Asynchronous Transfer Mode (ATM) pseudowire to be carried over a packet-switched network (PSN) in any mode. This table contains a set of parameters for the ATM pseudowire that needs to be adapted and carried over a PSN.

The jnxpwAtmCfgTable is indexed by pwIndex, which was created for the associated entry in the pwTable defined in PW-STD-MIB.

An jnxpwAtmCfgEntry object is created for every new ATM type associated pwType in the pwTable.



NOTE: Unless otherwise specified:

- All read-write objects in the jnxpwAtmCfgTable table might get changed when the pseudowire is defined as inactive.
- All read-write object values remain unchanged after a reboot.

Each JnxPwAtmCfgEntry, whose object identifier is {jnxpwAtmCfgTable 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex. Each JnxPwAtmCfgEntry contains the objects listed in [Table 307 on page 730](#).

Table 307: JnxPwAtmCfgEntry

Object	Object Identifier	Description
jnxpwAtmCfgMaxCellConcatenation	jnxpwAtmCfgEntry 1	Indicates the maximum number of ATM cells that can be concatenated into one pseudowire packet toward the PSN. NOTE: In non-LDP and other signaling protocol environments, this object might get changed, and therefore the traffic might get interrupted.
jnxpwAtmCfgFarEndMaxCellConcatenation	jnxpwAtmCfgEntry 2	Indicates the maximum number of ATM cells that can be concatenated into one pseudowire packet toward the PSN. This number is reported by the far end. If no LDP is in use, either this object returns the value 0 or allows you to set a value in it for calculating protocol overhead.

Table 307: JnxPwAtmCfgEntry (continued)

Object	Object Identifier	Description
jnxpwAtmCfgTimeoutMode	jnxpwAtmCfgEntry 3	<p>Determines whether or not a packet can be transmitted to the PSN based on time-out expiration for collecting cells.</p> <p>NOTE: The actual handling of the time-out is implementation specific—as such this object may be changed at any time under proper consideration of traffic interruption effect.</p>
jnxpwAtmClpQosMapping	jnxpwAtmCfgEntry 4	<p>Indicates whether the Cell Loss Priority (CLP) bits need to be considered when setting the value in the Quality of Service fields of the encapsulating protocol (for example, EXP fields of the MPLS Label Stack).</p> <p>The value of this object can be changed any time. However, the value of this object must always be set to false (2) in transparent cell transport mode.</p> <p>The drop precedence can be preserved across the PSN by setting this object to true (?).</p>

jnxpwAtmOutboundNto1Table

The jnxpwAtmOutboundNto1Table, whose object identifier is {jnxpwAtmObjects 2}, specifies the information required for an Asynchronous Transfer Mode (ATM) pseudowire to be carried over packet-switched network (PSN) in the outbound direction. The jnxpwAtmOutboundNto1Entry object represents an ATM pseudowire that needs to be adapted and carried over the PSN. This table is indexed by pwIndex from pwTable and the ATM interface is indexed with Virtual Path Links (VPLs) and Virtual Channel Links (VCLs).

For every entry in the pwTable with a pwType of atmCellNto1Vcc(9) or atmCellNto1Vpc(10), up to *N* entries can be created in the jnxpwAtmOutboundNto1Table table.

For every entry in the pwTable with a pwType of atmCelllto1Vcc(12), atmCelllto1Vpc(13), atmAal5PduVcc(14), atmAal5SduVcc(2), or atmTransparent(3), a single entry is created in the jnxpwAtmOutboundNto1Table table..

**NOTE:**

- An entry can be created only when the Virtual Paths (VP) and Virtual Channels (VC) is known.
- For pseudowire of type atmTransparent(3), Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) is 0xFFFF during the set operation.
- Unless otherwise specified, all read-create objects in the jnxpwAtmOutboundNto1Table table must not be changed after row activation. They should remain unchanged after reboot.

Each jnxpwAtmOutboundNto1Entry, whose object identifier is {jnxpwAtmOutboundNto1Table 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex. Each JnxPwAtmCfgEntry contains the objects listed in [Table 308 on page 732](#).

Table 308: jnxpwAtmOutboundNto1Entry

Object	Object Identifier	Description
jnxpwAtmOutboundNto1AtmIf	jnxpwAtmOutboundNto1Entry 1	Represents the ATM Interface that receives cells from the ATM network.
jnxpwAtmOutboundNto1Vpi	jnxpwAtmOutboundNto1Entry 2	Specifies the VPI value of the current ATM pseudowire. For pseudowire of type atmTransparent(3), VPI value is 0xFFFF.
jnxpwAtmOutboundNto1Vci	jnxpwAtmOutboundNto1Entry 3	Specifies the VCI value of the current ATM pseudowire. For pseudowire of type atmTransparent(3) or VP, VCI value is 0xFFFF.
jnxpwAtmOutboundNto1RowStatus	jnxpwAtmOutboundNto1Entry 4	Object used for creating, modifying, or deleting a row in the jnxpwAtmOutboundNto1Table.
jnxpwAtmOutboundNto1TrafficParamDescr	jnxpwAtmOutboundNto1Entry 5	Contains a pointer to an ATM traffic parameter-specific row in the private or standard table that will be used when receiving cells from the ATM network. NOTE: The jnxpwAtmOutboundNto1Table table contains a set of self-consistent ATM traffic parameters including the ATM traffic service category. The value 0.0 in the ATM traffic service category indicates <i>Best Effort</i> .

Table 308: jnxpwAtmOutboundNto1Entry (*continued*)

Object	Object Identifier	Description
jnxpwAtmOutboundNto1MappedVpi	jnxpwAtmOutboundNto1Entry 6	<p>Contains the egress generated VPI value of the current ATM pseudowire. This entry is valid for pseudowire of type atmCellNto1Vcc(9), atmCellNto1Vpc(10), atmCell1to1Vcc(12), or atmCell1to1Vpc(13). For other types, the value is 0xFFFF.</p> <p>NOTE: This value might change when the pseudowire is defined as inactive.</p>
jnxpwAtmOutboundNto1MappedVci	jnxpwAtmOutboundNto1Entry 7	<p>Contains the egress generated VCI value of the current ATM pseudowire. This entry is valid for pseudowire of type atmCellNto1Vcc(9), atmCellNto1Vpc(10), atmCell1to1Vcc(12), or atmCell1to1Vpc(13). For VP case and other types, the value is 0xFFFF.</p> <p>NOTE: This value might change when the pseudowire is defined as inactive.</p>



NOTE: Devices capable of implementing N:1, 1:1, and transparent cell transport modes support the N:1 table for all modes (with respective applicable settings). For such devices, you can create an entry for 1:1 or transparent cell transport modes in jnxpwAtmInboundNto1Table. The respective line in the jnxpwAtmOutboundNto1Table is automatically created.

jnxpwAtmInboundNto1Table

The jnxpwAtmInboundNto1Table, whose object identifier is {jnxpwAtmObjects 3}, specifies the information required for an Asynchronous Transfer Mode (ATM) pseudowire to be carried over packet-switched network (PSN) in the outbound direction. An jnxpwAtmInboundNto1Entry object represents the ATM pseudowire that needs to be adapted and carried over the PSN. This table is indexed by pwIndex from pwTable and the ATM interface is indexed with Virtual Path Links (VPLs) and Virtual Channel Links (VCLs).

For every entry in the pwTable with a pwType of atmCellNto1Vcc(9) or atmCellNto1Vpc(10), up to *N* entries can be created in the jnxpwAtmInboundNto1Table table.

For every entry in the pwTable with a pwType of atmCell1to1Vcc(12), atmCell1to1Vpc(13), atmAal5PduVcc(14), atmAal5SduVcc(2), or atmTransparent(3), a single entry is created in the jnxpwAtmInboundNto1Table.

**NOTE:**

- An entry can be created only when the Virtual Paths (VP) and Virtual Channels (VC) is known.
- For pseudowire of type atmTransparent(3), Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) is 0xFFFF during the set operation.
- Unless otherwise specified, all read-create objects in the jnxpwAtmInboundNto1Table must not be changed after row activation. They should remain unchanged after reboot.

Each jnxpwAtmInboundNto1Entry, whose object identifier is {jnxpwAtmInboundNto1Table 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex. Each JnxPwAtmCfgEntry contains the objects listed in [Table 309 on page 734](#).

Table 309: jnxpwAtmInboundNto1Entry

Object	Object Identifier	Description
jnxpwAtmInboundNto1AtmIf	jnxpwAtmInboundNto1Entry 1	Represents the ATM Interface that receives cells from the ATM network.
jnxpwAtmInboundNto1Vpi	jnxpwAtmInboundNto1Entry 2	Specifies the VPI value of the current ATM pseudowire. For pseudowire of type atmTransparent(3), the VPI value is 0xFFFF.
jnxpwAtmInboundNto1Vci	jnxpwAtmInboundNto1Entry 3	Specifies the VCI value of the current ATM pseudowire. For pseudowire of type atmTransparent(3) or VP, the VCI value is 0xFFFF.
jnxpwAtmInboundNto1RowStatus	jnxpwAtmInboundNto1Entry 4	Object used for creating, modifying, or deleting a row in the jnxpwAtmInboundNto1Table.
jnxpwAtmInboundNto1TrafficParamDescr	jnxpwAtmInboundNto1Entry 5	Contains a pointer to an ATM traffic parameter-specific row in the private or standard table that will be used when receiving cells from the ATM network. NOTE: The jnxpwAtmInboundNto1Table contains a set of self-consistent ATM traffic parameters including the ATM traffic service category. The value 0.0 in the ATM traffic service category indicates <i>Best Effort</i> .

Table 309: jnxpwAtmInboundNto1Entry (*continued*)

Object	Object Identifier	Description
jnxpwAtmInboundNto1MappedVpi	jnxpwAtmInboundNto1Entry 6	<p>Contains the generated VPI value of the current ATM pseudowire. This entry is valid for pseudowire of type atmCellNto1Vcc(9), atmCellNto1Vpc(10), atmCell1to1Vcc(12), or atmCell1to1Vpc(13). For other types, the value is 0xFFFF.</p> <p>NOTE: This value might change when the pseudowire is defined as inactive.</p>
jnxpwAtmInboundNto1MappedVci	jnxpwAtmInboundNto1Entry 7	<p>Contains the generated VCI value of the current ATM pseudowire. This entry is valid for pseudowire of type atmCellNto1Vcc(9), atmCellNto1Vpc(10), atmCell1to1Vcc(12), or atmCell1to1Vpc(13). For VP and other types, the value is 0xFFFF.</p> <p>NOTE: This value might change when the pseudowire is defined as inactive.</p>



NOTE: Devices capable of implementing N:1, 1:1, and transparent cell transport modes support the N:1 table for all modes (with respective applicable settings). For such devices, you can create an entry for 1:1 or transparent cell transport modes in jnxpwAtmInboundNto1Table. The respective line in the jnxpwAtmOutboundNto1Table is automatically created.

jnxpwAtmPerfCurrentTable

The jnxpwAtmPerfCurrentTable, whose object identifier is {jnxpwAtmObjects 4}, contains performance information for each Asynchronous Transfer Mode (ATM) pseudowire for the current 15-minute interval.

A jnxpwAtmPerfCurrentEntry is created for each jnxpwAtmPerfCurrentTable entry. When the 15-minute interval expires, the contents of the jnxpwAtmPerfCurrentEntry are transferred to jnxpwAtmPerfCurrentTable, and the jnxpwAtmPerfCurrentEntry counters are reset.

Each jnxpwAtmPerfCurrentEntry, whose object identifier is {jnxpwAtmPerfCurrentTable 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, and jnxVpnPwIndex. Each jnxpwAtmPerfCurrentEntry contains the objects listed in [Table 310 on page 735](#).

Table 310: jnxpwAtmPerfCurrentEntry

Object	Object ID	Description
jnxpwAtmPerfCurrentMissingPkts	jnxpwAtmPerfCurrentEntry 1	Indicates the number of missing packets as detected from the control word sequence number gaps.

Table 310: jnxpwAtmPerfCurrentEntry (*continued*)

Object	Object ID	Description
jnxpwAtmPerfCurrentPktsReOrder	jnxpwAtmPerfCurrentEntry 2	Indicates the number of packets detected out of sequence (based on control word sequence number), but successfully reordered. NOTE: Some implementations may not support this feature. Therefore, we recommend you to test this feature in your networking environment before implementing it.
jnxpwAtmPerfCurrentPktsMisOrder	jnxpwAtmPerfCurrentEntry 3	Indicates the number of packets detected as out of order (based on control word sequence numbers).
jnxpwAtmPerfCurrentPktsTimeout	jnxpwAtmPerfCurrentEntry 4	Indicates the number of packets transmitted because of time-out expiration when attempting to collect cells.
jnxpwAtmPerfCurrentPktsXmit	jnxpwAtmPerfCurrentEntry 5	Indicates the number of transmitted packets.
jnxpwAtmPerfCurrentCellsDropped	jnxpwAtmPerfCurrentEntry 6	Indicates the number of dropped cells.
jnxpwAtmPerfCurrentPktsReceived	jnxpwAtmPerfCurrentEntry 7	Indicates the number of received packets.
jnxpwAtmPerfCurrentUnknownCells	jnxpwAtmPerfCurrentEntry 8	Indicates the number of cells received from the packet-switched network (PSN) with unknown Virtual Path Identifier (VPI) or Virtual Channel Identifier (VCI) values. NOTE: This object is relevant only in N:1 mode.

jnxpwAtmPerfIntervalTable

The jnxpwAtmPerfIntervalTable, whose object identifier is {jnxpwAtmObjects 5}, provides performance information for each ATM pseudowire for the past 15-minute intervals. Typically, this table has a maximum of 96 entries for a 24-hour period. However, the number of entries might vary in some cases. For each 15-minute interval, a jnxpwAtmPerfIntervalEntry is created in this table for every jnxpwAtmPerfCurrentEntry that is 15 minutes old. The contents of the current jnxpwAtmPerfCurrentEntry are copied to the new jnxpwAtmPerfIntervalEntry in this table. The current jnxpwAtmPerfCurrentEntry then resets its counts to zero for the next current 15-minute interval.

Each jnxpwAtmPerfIntervalEntry, whose object identifier is {jnxpwAtmPerfIntervalTable 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, jnxVpnPwIndex, and jnxpwAtmPerfIntervalNumber. Each jnxpwAtmPerfIntervalEntry contains the objects listed in [Table 311 on page 737](#).

Table 311: jnxpwAtmPerfIntervalEntry

Object	Object ID	Description
jnxpwAtmPerfIntervalNumber	jnxpwAtmPerfIntervalEntry 1	<p>Identifies the interval for which the statistics are available. The range of numbers is 1 through 96 for a 24-hour period.</p> <p>The interval identified by 1 is the most recently completed 15-minute interval, and the interval identified by N is the interval immediately preceding the one identified by N-1. The minimum range of N is 1 through 4. The default range is 1 through 32.</p>
jnxpwAtmPerfIntervalValidData	jnxpwAtmPerfIntervalEntry 2	Indicates whether the data for the interval is valid.
jnxpwAtmPerfIntervalDuration	jnxpwAtmPerfIntervalEntry 3	<p>Indicates the duration of a particular interval in seconds.</p> <p>NOTE: Adjustments in the system's time-of-day clock might cause the interval to be greater or less than the actual value. Therefore this actual interval value is provided.</p>
jnxpwAtmPerfIntervalMissingPkts	jnxpwAtmPerfIntervalEntry 4	Indicates the number of missing packets as detected from the control word sequence number gaps.
jnxpwAtmPerfIntervalPktsReOrder	jnxpwAtmPerfIntervalEntry 5	<p>Indicates the number of packets detected out of sequence (based on control word sequence number), but successfully reordered.</p> <p>NOTE: Some implementations may not support this feature. Therefore, we recommend you to test this feature in your networking environment before implementing it.</p>
jnxpwAtmPerfIntervalPktsMisOrder	jnxpwAtmPerfIntervalEntry 6	Indicates the number of packets detected as out of order (based on the control word sequence numbers).
jnxpwAtmPerfIntervalPktsTimeout	jnxpwAtmPerfIntervalEntry 7	Indicates the number of packets transmitted because of time-out expiration.
jnxpwAtmPerfIntervalPktsXmit	jnxpwAtmPerfIntervalEntry 8	Indicates the number of transmitted packets.
jnxpwAtmPerfIntervalCellsDropped	jnxpwAtmPerfIntervalEntry 9	Indicates the number of dropped cells.

Table 311: jnxpwAtmPerfIntervalEntry (*continued*)

Object	Object ID	Description
jnxpwAtmPerfIntervalPktsReceived	jnxpwAtmPerfIntervalEntry 10	Indicates the number of received packets.
jnxpwAtmPerfIntervalUnknownCells	jnxpwAtmPerfIntervalEntry 11	Indicates the number of cells received from the packet-switched network (PSN) with unknown Virtual Path Identifier (VPI) or Virtual Channel Identifier (VCI) values. NOTE: This object is relevant only in N:1 mode.

jnxpwAtmPerf1DayIntervalTable

The jnxpwAtmPerf1DayIntervalTable, whose object identifier is {jnxpwAtmObjects 6}, provides performance information for each ATM pseudowire for the past one-day intervals up to one full month. For each day, one jnxpwAtmPerf1DayIntervalEntry is created in this table.

Each jnxpwAtmPerf1DayIntervalEntry, whose object identifier is {jnxpwAtmPerf1DayIntervalTable 1}, is indexed by a combination of jnxVpnPwVpnType, jnxVpnPwVpnName, jnxVpnPwIndex, and jnxpwAtmPerf1DayIntervalNumber. Each jnxpwAtmPerf1DayIntervalEntry contains the objects listed in [Table 312 on page 738](#).

Table 312: jnxpwAtmPerf1DayIntervalEntry

Object	Object Identifier	Description
jnxpwAtmPerf1DayIntervalNumber	jnxpwAtmPerf1DayIntervalEntry 1	Indicates interval period. The value 1 indicates current day and 2 and above indicate previous days respectively
jnxpwAtmPerf1DayIntervalValidData	jnxpwAtmPerf1DayIntervalEntry 2	Indicates whether the data for the interval is valid.
jnxpwAtmPerf1DayIntervalDuration	jnxpwAtmPerf1DayIntervalEntry 3	Indicates the duration of a particular interval in seconds. NOTE: Adjustments in the system's time-of-day clock might cause the interval to be greater or less than the actual value. Therefore this actual interval value is provided.
jnxpwAtmPerf1DayIntervalMissingPkts	jnxpwAtmPerf1DayIntervalEntry 4	Indicates the number of missing packets as detected from the control word sequence number gaps.

Table 312: jnxpwAtmPerf1DayIntervalEntry (*continued*)

Object	Object Identifier	Description
jnxpwAtmPerf1DayIntervalPktsReOrder	jnxpwAtmPerf1DayIntervalEntry 5	Indicates the number of packets detected out of sequence (based on control word sequence number), but successfully reordered. NOTE: Some implementations may not support this feature. Therefore, we recommend you to test this feature in your networking environment before implementing it.
jnxpwAtmPerf1DayIntervalPktsMisOrder	jnxpwAtmPerf1DayIntervalEntry 6	Indicates the number of packets detected as out of order (based on the control word sequence numbers), and could not be reordered.
jnxpwAtmPerf1DayIntervalPktsTimeout	jnxpwAtmPerf1DayIntervalEntry 7	Indicates the number of packets transmitted because of time-out expiration.
jnxpwAtmPerf1DayIntervalPktsXmit	jnxpwAtmPerf1DayIntervalEntry 8	Indicates the number of transmitted packets.
jnxpwAtmPerf1DayIntervalCellsDropped	jnxpwAtmPerf1DayIntervalEntry 9	Indicates the number of dropped cells.
jnxpwAtmPerf1DayIntervalPktsReceived	jnxpwAtmPerf1DayIntervalEntry 10	Indicates the number of received packets.
jnxpwAtmPerf1DayIntervalUnknownCells	jnxpwAtmPerf1DayIntervalEntry 11	Indicates the number of cells received from the packet-switched network (PSN) with unknown Virtual Path Identifier (VPI) or Virtual Channel Identifier (VCI) values. NOTE: This object is relevant only in N:1 mode.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 62

Pseudowire TDM MIB

- [Interpreting the Enterprise-Specific Pseudowire TDM MIB on page 741](#)

Interpreting the Enterprise-Specific Pseudowire TDM MIB

- [Pseudowire TDM MIB on page 741](#)
- [Pseudowire TDM MIB Textual Conventions on page 742](#)
- [jnxpwTDMTable on page 742](#)
- [jnxpwTDMCfgTable on page 744](#)
- [jnxpwTDMPerfCurrentTable on page 746](#)
- [jnxpwTDMPerfIntervalTable on page 747](#)
- [jnxpwTDMPerf1DayIntervalTable on page 748](#)

Pseudowire TDM MIB

The enterprise-specific Pseudowire TDM MIB is the Juniper Networks implementation of the standard *Managed Objects for Time-Division Multiplexing (TDM) over Packet Switched Network MIB* (internet draft draft-ietf-pwe3-tdm-mib-08.txt).

The Juniper Networks enterprise-specific Pseudowire TDM MIB contains information about configuration and statistics for specific pseudowire types. The enterprise-specific MIB also contains definitions of managed objects for encapsulating TDM (T1, E1, E3, and NxDS0) as pseudowire over packet switched networks.

For a downloadable version of this MIB, see .

Starting with Junos OS Release 12.2, the Pseudowire TDM MIB supports Circuit Emulation Service over Packet-Switched Network (CESoPSN) and Structure-Agnostic TDM over Packet (SATO P) encapsulations configured on Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP (MIC-3D-4COC3-1COC12-CE). The SAToP encapsulation is supported on T1 and E1 interfaces configured on this MIC. The CESoPSN encapsulation is supported on NxDS0 interfaces configured on the MIC.

For information about the Pseudowire TDM MIB, see the following topics:

- [Pseudowire TDM MIB Textual Conventions on page 742](#)
- [jnxpwTDMTable on page 742](#)

- [jnxpwTDMCfgTable](#) on page 744
- [jnxpwTDMPerfCurrentTable](#) on page 746
- [jnxpwTDMPerfIntervalTable](#) on page 747
- [jnxpwTDMPerf1DayIntervalTable](#) on page 748

Pseudowire TDM MIB Textual Conventions

The Juniper Networks enterprise-specific Pseudowire TDM MIB uses the following textual conventions:

- **jnxPwTDMCfgIndex**—Index of the relevant **pwXXxCfgTable**. The range of values is 1 through 4294967295.
- **jnxPwCfgIndexOrzero**—Index of the relevant configuration table for supplemental information regarding configuration of the specific technology. A value of 0 indicates that no additional configuration information is applicable. The range of values is 0 through 4294967295.

jnxpwTDMTable

The **jnxpwTDMTable**, whose object identifier is **{jnxpwTDMObjects 1}**, contains basic information including **ifIndex**, and pointers to entries in the relevant TDM configuration tables for a given TDM pseudowire.

The **jnxpwTDMTable** is indexed by **pwIndex**, which was created for the associated entry in the **pwTable** defined in **PW-STD-MIB**.

A **jnxpwTDMEntry** is created by the agent for each entry in the **pwTable** when the **pwType** is equal to one of the following:

- **e1Satop(17)**
- **t1Satop(18)**
- **e3Satop(19)**
- **t3Satop(20)**
- **basicCesPsn(21)**
- **basicTdmlp(22)**
- **tdmCasCesPsn(23)**
- **tdmCasTdmlp(24)**



NOTE: Unless otherwise specified, objects in this table should not be changed after the row activation, and should remain unchanged after a reboot.

Each `jnxpwTDMEntry`, whose object identifier is `{jnxpwTDMTable 1}`, is indexed by a combination of `jnxVpnPwVpnType`, `jnxVpnPwVpnName`, and `jnxVpnPwIndex`. Each `jnxpwTDMEntry` contains the objects listed in [Table 313 on page 743](#).

Table 313: `jnxpwTDMEntry`

Object	Object Identifier	Description
<code>jnxpwTDMRate</code>	<code>jnxpwTDMEntry 1</code>	<p>Represents the bit-rate of the TDM service in multiples of 64 Kbps.</p> <p>This object complements the definition of the <code>pwType</code> object in the <code>PW-STD-MIB</code>.</p> <p>For structure-agnostic emulations, one of the following values is used:</p> <ul style="list-style-type: none"> • 24—Satop T1 basic emulation • 25—Satop T1 octet-aligned emulation • 32—Satop E1 • 535—Satop E3 • 699—Satop T3 <p>For structure-aware emulations, the value of this object must be set to N, where N denotes the number of DS0 channels in the corresponding attachment circuit.</p>
<code>jnxpwTDMIfIndex</code>	<code>jnxpwTDMEntry 2</code>	<p>Contains a unique index within the <code>ifTable</code>. This object represents the interface index of the full link or the interface index for the bundle holding the group of time slots to be transmitted through the pseudowire connection.</p> <p>A value of zero indicates an interface index that is to be determined.</p> <p>When a configured TDM <code>ifIndex</code> is removed, the agent deletes the associated pseudowire rows. If the agent does not delete the rows, the agent must set this object to zero.</p>
<code>jnxpwTDMTimeElapsed</code>	<code>jnxpwTDMEntry 6</code>	<p>Indicates the time, in seconds, since the beginning of the current measurement period. If the current interval exceeds the maximum value, the agent returns the maximum value.</p>
<code>jnxpwTDMValidIntervals</code>	<code>jnxpwTDMEntry 7</code>	<p>Indicates the number of previous 15-minute intervals for which data was collected. An agent with TDM capability must be able to support 4 through 96 intervals. The default is 32.</p>
<code>jnxpwTDMValidDayIntervals</code>	<code>jnxpwTDMEntry 8</code>	<p>Indicates the number of previous days for which the data was collected. An agent with TDM capability supports 1 through 30 days. The default value is 1.</p>

jnxpwTDMCfgTable

The **jnxpwTDMCfgTable**, whose object identifier is **{jnxpwTDMObjects 3}**, contains **jnxpwTDMCfgEntry** objects that define the characteristics of a TDM pseudowire.

Entries created in the **jnxpwTDMCfgTable** are reused by multiple pseudowires. Unless otherwise specified, entries created in this table should not be modified after the row activation. If the row index is shared with any entry in the **pwTDMTable**, rows should remain unchanged after reboot.

Each **jnxpwTDMCfgEntry**, whose object identifier is **{jnxpwTDMCfgTable 1}**, contains the objects listed in [Table 314 on page 744](#).

Table 314: jnxpwTDMCfgEntry

Object	Object Identifier	Description
jnxpwTDMCfgRowStatus	jnxpwTDMCfgEntry 2	Contains the control object that is used for creating, modifying, and deleting a row from this table. The following objects should not be modified if the entry is in use and the status is active: pwTDMCfgPayloadSize , pwTDMCfgRtpHdrUsed , pwTDMCfgJtrBfrDepth , and pwTDMCfgPayloadSuppression . The row should not be deleted if the entry is in use.
jnxpwTDMCfgPayloadSize	jnxpwTDMCfgEntry 4	Indicates the payload size (in bytes) to be defined during the pseudowire setup. Upon transmit, the implementation must be capable of carrying the payload indicated here. Upon receive, when the LEN field is set to 0, the payload of the packet assumes this size, and if the actual packet size is inconsistent with this length, the packet is considered to be malformed.
jnxpwTDMCfgRtpHdrUsed	jnxpwTDMCfgEntry 6	Indicates whether the RTP header is prepended to a TDM packet. If set to True , the RTP header is prepended to the TDM packet. If set to False , the RTP header is not prepended.
jnxpwTDMCfgJtrBfrDepth	jnxpwTDMCfgEntry 7	Indicates the size of the buffer. The size of the buffer is locally configured to accommodate the PSN-specific packet delay variation. If configured to a value that is not supported by the implementation, the agent returns an error code jtrBfrDepth in pwTDMConfigError . NOTE: Because jitter buffers are a limited resource to be managed, the actual size must be at least twice as big as the value of pwTDMCfgJtrBfrDepth .

Table 314: jnxpwTDMCfEntry (continued)

Object	Object Identifier	Description
jnxpwTDMCfConsecPktsInSynch	jnxpwTDMCfEntry 9	Denotes the number of consecutive packets with sequential numbers that are required to exit the LOPS state. The value of this object might change when the related pseudowire is configured as not active.
jnxpwTDMCfConsecMissPktsOutSynch	jnxpwTDMCfEntry 10	Denotes the number of consecutive missing packets that are required to enter the LOPS state. The value of this object might change when the related pseudowire is configured as not active.
jnxpwTDMCfSetUp2SynchTimeOut	jnxpwTDMCfEntry 11	<p>Denotes the amount of time the host should wait before declaring a pseudowire as down when the number of consecutive TDM packets that have been received after changing the administrative status to up and after finalization of signaling (if supported) between the two PEs is fewer than pwTDMCfConsecPktsInSynch. When the pseudowire has OperStatus as up, this parameter is not valid. This parameter is defined to prevent the host from prematurely declaring a pseudowire as down. Pseudowire down notifications cannot be sent before this timer expires.</p> <p>This parameter is valid only when administrative changes cause pseudowire status changes. If the pseudowire fails due to network problems, a down notification must be sent.</p> <p>The value of this object might change when the related pseudowire is configured as not active.</p>
jnxpwTDMCfPktReplacePolicy	jnxpwTDMCfEntry 12	Indicates the value to be played when CE-bound packets have jitter buffer over or under flow, or are missing for any reason. The value of this object might change when the related pseudowire is configured as not active.
jnxpwTDMCfAvePktLossTimeWindow	jnxpwTDMCfEntry 13	Indicates the time interval over which the average packet loss rate is calculated to detect excessive packet loss rate. The value of this object might change when the related pseudowire is configured as not active.

Table 314: jnxpwTDMCfgEntry (continued)

Object	Object Identifier	Description
jnxpwTDMCfgExcessivePktLossThreshold	jnxpwTDMCfgEntry 14	<p>Indicates the threshold value based on which excessive packet loss rate is detected. Excessive packet loss is detected by computing the average packet loss rate over pwTDMCfgAvePktLossTimeWindow and comparing it with the threshold value. The rate is expressed as a percentage.</p> <p>The value of this object might change when the related pseudowire is configured as not active.</p>
jnxpwTDMCfgAlarmThreshold	jnxpwTDMCfgEntry 15	<p>The amount of time before an alarm is reported. Only those defect states that persist for the length of time specified by this object trigger an alarm notification. The time interval is counted in milliseconds.</p> <p>The value of this object might change when the related pseudowire is configured as not active.</p>
jnxpwTDMCfgClearAlarmThreshold	jnxpwTDMCfgEntry 16	<p>Indicates the amount of time before clearing an alarm. If the corresponding defect is not detected through the length of time specified through this object, the alarm is cleared. The amount of time is measured in milliseconds.</p> <p>The value of this object might change when the related pseudowire is configured as not active.</p>
jnxpwTDMCfgStorageType	jnxpwTDMCfgEntry 19	Indicates the storage type for this row.
jnxpwTDMCfgPktFiller	jnxpwTDMCfgEntry 20	<p>Denotes the filler byte pattern played out on the TDM interface when pwTDMCfgPktReplacePolicy is set to filler(3).</p> <p>The value of this object might change when the related pseudowire is configured as not active.</p>

jnxpwTDMPerfCurrentTable

The **jnxpwTDMPerfCurrentTable**, whose object identifier is **{jnxpwTDMObjects 5}**, contains information about pseudowire performance per TDM for the current 15-minute interval.

A **jnxpwTDMPerfCurrentEntry** is created for each **pwTDMTable** entry. When the 15-minute interval expires, the contents of the **jnxpwTDMPerfCurrentEntry** are transferred to **pwTDMPerfIntervalTable**, and the **jnxpwTDMPerfCurrentEntry** counters are reset.

Each `jnxpwTDMPerfCurrentEntry`, whose object identifier is `{jnxpwTDMPerfCurrentTable 1}`, contains the objects listed in [Table 315 on page 747](#).

Table 315: jnxpwTDMPerfCurrentEntry

Object	Object ID	Description
<code>jnxpwTDMPerfCurrentMissingPkts</code>	<code>jnxpwTDMPerfCurrentEntry 1</code>	Indicates the number of missing packets as detected from the control word sequence number gaps.
<code>jnxpwTDMPerfCurrentJtrBfrUnderruns</code>	<code>jnxpwTDMPerfCurrentEntry 3</code>	Indicates the number of times a packet had to be played out while the jitter buffer was empty.
<code>jnxpwTDMPerfCurrentMisOrderDropped</code>	<code>jnxpwTDMPerfCurrentEntry 4</code>	Indicates the number of packets detected out of order (based on control word sequence numbers), and could neither be reordered nor fit in the jitter buffer.
<code>jnxpwTDMPerfCurrentMalformedPkt</code>	<code>jnxpwTDMPerfCurrentEntry 5</code>	Indicates the number of packets that are detected with unexpected size or bad header stack.
<code>jnxpwTDMPerfCurrentESs</code>	<code>jnxpwTDMPerfCurrentEntry 6</code>	Denotes the counter associated with the number of Error Seconds encountered. Any malformed packet, sequence error, LOPS, and so on are considered as an Error Second.
<code>jnxpwTDMPerfCurrentSESs</code>	<code>jnxpwTDMPerfCurrentEntry 7</code>	Denotes the counter associated with the number of Severely Error Seconds (SES) encountered.
<code>jnxpwTDMPerfCurrentUASs</code>	<code>jnxpwTDMPerfCurrentEntry 8</code>	Denotes the counter associated with the number of Unavailable Seconds (UAS) encountered. Any consecutive 10 seconds of SES are counted as one UAS.

jnxpwTDMPerfIntervalTable

The `jnxpwTDMPerfIntervalTable`, whose object identifier is `{jnxpwTDMObjects 6}`, provides historical performance information per TDM pseudowire for 15-minute intervals. Typically, this table has a maximum of 96 entries for a 24-hour period. However, the number of entries might vary in some cases. For each 15-minute interval, a `jnxpwTDMPerfIntervalEntry` is created.

Each `jnxpwTDMPerfIntervalEntry` contains the objects listed in [Table 316 on page 748](#).

Table 316: jnxpwTDMPerfIntervalEntry

Object	Object ID	Description
jnxpwTDMPerfIntervalNumber	jnxpwTDMPerfIntervalEntry 1	Identifies the interval for which the statistics are available. The range of numbers is 1 through 96 for a 24-hour period. The interval identified by 1 is the most recently completed 15-minute interval, and the interval identified by N is the interval immediately preceding the one identified by N-1 . The minimum range of N is 1 through 4. The default range is 1 through 32.
jnxpwTDMPerfIntervalValidData	jnxpwTDMPerfIntervalEntry 2	Indicates whether the data for the given interval is valid.
jnxpwTDMPerfIntervalMissingPkts	jnxpwTDMPerfIntervalEntry 4	Contains the number of missing packets (as detected from the control word sequence number gaps).
jnxpwTDMPerfIntervalJtrBfrUnderruns	jnxpwTDMPerfIntervalEntry 6	Indicates the number of times a packet had to be played out and the jitter buffer was empty.
jnxpwTDMPerfIntervalMisOrderDropped	jnxpwTDMPerfIntervalEntry 7	Indicates the number of packets detected as out of order, based on the control word sequence numbers, and could not be reordered or could not fit in the jitter buffer.
jnxpwTDMPerfIntervalMalformedPkt	jnxpwTDMPerfIntervalEntry 8	Indicates the number of packets detected with unexpected size or bad headers' stack.
jnxpwTDMPerfIntervalESs	jnxpwTDMPerfIntervalEntry 9	Denotes the counter that is associated with the number of Error Seconds encountered.
jnxpwTDMPerfIntervalSESs	jnxpwTDMPerfIntervalEntry 10	Denotes the counter associated with the number of Severely Error Seconds (SES) encountered.
jnxpwTDMPerfIntervalUASs	jnxpwTDMPerfIntervalEntry 11	Denotes the counter associated with the number of Unavailable Seconds (UAS) encountered.

jnxpwTDMPerf1DayIntervalTable

The **jnxpwTDMPerf1DayIntervalTable** contains performance information per TDM pseudowire similar to the information in the **pwTDMPerfIntervalTable**. However, unlike the **pwTDMPerfIntervalTable** that contains historical data for 15-minute intervals, this

table contains data for one-day intervals for up to a month. For each day, one `jnxpwTDMPerf1DayIntervalEntry` is created in the table.

Each `jnxpwTDMPerf1DayIntervalEntry`, whose object identifier is `{jnxpwTDMPerf1DayIntervalTable 1}`, contains the objects listed in [Table 317 on page 749](#).

Table 317: jnxpwTDMPerf1DayIntervalEntry

Object	Object Identifier	Description
<code>jnxpwTDMPerf1DayIntervalValidData</code>	<code>jnxpwTDMPerf1DayIntervalEntry 2</code>	Indicates whether the data for this interval is valid.
<code>jnxpwTDMPerf1DayIntervalMissingPkts</code>	<code>jnxpwTDMPerf1DayIntervalEntry 4</code>	Indicates the number of missing packets as detected from the control word sequence number gaps.
<code>jnxpwTDMPerf1DayIntervalJtrBfrUnderruns</code>	<code>jnxpwTDMPerf1DayIntervalEntry 6</code>	Indicates the number of times a packet had to be played out and the jitter buffer was empty.
<code>jnxpwTDMPerf1DayIntervalMisOrderDropped</code>	<code>jnxpwTDMPerf1DayIntervalEntry 7</code>	Indicates the number of packets that are detected as out of order (based on the control word sequence numbers), and could not be reordered or could not fit in the jitter buffer.
<code>jnxpwTDMPerf1DayIntervalMalformedPkt</code>	<code>jnxpwTDMPerf1DayIntervalEntry 8</code>	Indicates the number of packets that are detected with unexpected size, or bad headers' stack.
<code>jnxpwTDMPerf1DayIntervalESs</code>	<code>jnxpwTDMPerf1DayIntervalEntry 9</code>	Denotes the counter associated with the number of Error Seconds encountered.
<code>jnxpwTDMPerf1DayIntervalSESSs</code>	<code>jnxpwTDMPerf1DayIntervalEntry 10</code>	Denotes the counter associated with the number of Severely Error Seconds (SES).
<code>jnxpwTDMPerf1DayIntervalUASs</code>	<code>jnxpwTDMPerf1DayIntervalEntry 11</code>	Denotes the counter associated with the number of Unavailable Seconds (UAS). When first entering the UAS state, the number of SES To UAS is added to this object. Then, as each additional UAS occurs, this object increments by one.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 63

Real-Time Performance Monitoring MIB

- Interpreting the Enterprise-Specific Real-Time Performance Monitoring MIB on page 751

Interpreting the Enterprise-Specific Real-Time Performance Monitoring MIB

- Real-Time Performance Monitoring MIB on page 751
- jnxRpmResultsSampleTable on page 752
- JnxRpmTimestampType on page 754
- jnxRpmResultsSummaryTable on page 754
- jnxRpmResultsCalculatedTable on page 755
- jnxRpmHistorySampleTable on page 756
- jnxRpmHistorySummaryTable on page 756
- jnxRpmHistoryCalculatedTable on page 757

Real-Time Performance Monitoring MIB

The Juniper Networks enterprise-specific Real-Time Performance Monitoring (RPM) MIB enables you to access real-time performance-related data over SNMP. Starting with Junos OS Release 8.4, you can access jitter measurements and calculations over SNMP.

The RPM MIB represents a restructuring of the standard Ping MIB and converts the flat structure of the Ping MIB into a hierarchical collection of data. For more information about Ping MIB, see Chapter 25, “Interpreting the Enterprise-Specific Ping MIB.” Similar to the Ping MIB, the RPM MIB also has two groups of tables: the Results group and the History group. The RPM MIB, however, groups its data into separate collection types and measurement sets.

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rpm.txt.

For information about the enterprise-specific RPM MIB objects, see the following topics:

- jnxRpmResultsSampleTable on page 752
- JnxRpmTimestampType on page 754
- jnxRpmResultsSummaryTable on page 754
- jnxRpmResultsCalculatedTable on page 755

- [jnxRpmHistorySampleTable](#) on page 756
- [jnxRpmHistorySummaryTable](#) on page 756
- [jnxRpmHistoryCalculatedTable](#) on page 757

jnxRpmResultsSampleTable

The **jnxRpmResultsSampleTable** provides you measurements from the latest individual RPM probe samples. Each **jnxRpmResultsSampleEntry** has the objects listed in [Table 318 on page 752](#).



NOTE: **jnxRpmResultsSampleTable** does not maintain entries for unsuccessful probes.

Table 318: jnxRpmResultsSampleTable

Object	Object Identifier	Description
jnxRpmResSampleType	jnxRpmResultsSampleEntry 1	The measurement type for the particular jnxRpmResultsSampleEntry .
jnxRpmResSampleValue	jnxRpmResultsSampleEntry 2	The measurement for the entry.
jnxRpmResSampleTsType	jnxRpmResultsSampleEntry 3	The type of timestamp used to obtain the measurement.
jnxRpmResSampleDate	jnxRpmResultsSampleEntry 4	The date and time at which the measurement was obtained.

For information about **JnxRpmMeasurementType**, see the following section:

- [JnxRpmMeasurementType](#) on page 752

JnxRpmMeasurementType

Different types of measurements can be made for each probe. [Table 319 on page 752](#) lists the measurement types used in **jnxRpm.mib**.

Table 319: jnxRpmMeasurementType

Measurement Type	Description
roundTripTime	The delay between the transmission of a probe and the arrival of its response.
rttJitter	The difference between the current round-trip time measurement and the previous one.

Table 319: jnxRpmMeasurementType (*continued*)

Measurement Type	Description
rttInterarrivalJitter	<p>An estimate of the statistical variance of a packet's interarrival time defined in RFC 1889 as:</p> $J = J + (D(i-1, i) - J) / 16$ <p>where J is the interarrival jitter and D (i-1, i) is the egressJitter measurement.</p>
egress	The delay between the transmission of a probe and its arrival at the destination.
egressJitter	The difference between the current egress delay and the previous measurement.
egressInterarrivalJitter	<p>An estimate of the statistical variance of a packet's interarrival time. Defined in RFC 1889 as:</p> $J = J + (D(i-1, i) - J) / 16$ <p>where J is the interarrival jitter and D (i-1, i) is the egressJitter measurement</p>
ingress	The delay between the transmission of a probe response and its arrival at the destination.
ingressJitter	The difference between the current ingress delay and the previous measurement.
ingressInterarrivalJitter	<p>An estimate of the statistical variance of a packet's interarrival time. Defined in RFC 1889 as:</p> $J = J + (D(i-1, i) - J) / 16$ <p>where J is the interarrival jitter and D (i-1, i) is the current ingressJitter measurement.</p>

Not all types of measurements are performed for every probe. For example, the jitter measurements are available only for those RPM entries that use hardware timestamps on both client and server. Similarly, the ingress and egress measurements are available only for those probe types that measure one-way delays or where hardware timestamps are used (for this, the one-way-hardware timestamp knob must be enabled). However, in these cases, if the one-way delay is greater than the round-trip time, the corresponding entries are not stored.



NOTE: To avoid possible variations in one-way jitter measurements and calculations due to clock synchronization issues, one-way jitter measurements are performed only on samples that are less than 10 seconds apart.

JnxRpmTimestampType

The following three types of timestamps are used to obtain measurements:

- **software**—Indicates that software-based timestamps are used on both client and server.
- **clientHardware**—Indicates that hardware-based timestamps are used on the client.
- **clientAndServerHardware**—Indicates that hardware-based timestamps are used on the RPM client and the server.

jnxRpmResultsSummaryTable

The **jnxRpmResultsSummaryTable** provides a summary of the results for each RPM entry (identified by **pingCtlOwnerIndex/pingCtlTestName** in the Ping MIB) and for each data collection maintained by that entry. The RPM feature maintains several different collections of probe data, providing overall summaries as well as detailed calculations for each collection.

The **jnxRpmResultsSummaryTable** maintains the following collection types:

- **currentTest**—The test that is being executed currently.
- **lastCompletedTest**—The most recently completed test.
- **movingAverage**—A list of most recent probes. You can configure the number of probes for this list using **jnxPingCtlMovAvgSize** or the **moving-average-size** command-line interface (CLI) command.
- **allTests**—All the probes that were sent. The value gets reset when the 64-bit value storing the square rolls over.

For each collection type, the table provides the following details:

- Number of probes sent
- Number of probes received
- Percentage of probes lost
- Timestamp for the latest sample in the collection

The **jnxRpmResultsSummaryEntry** has the objects listed in [Table 320 on page 754](#).

Table 320: jnxRpmResultsSummaryTable

Object	Object Identifier	Description
jnxRpmResSumCollection	jnxRpmResultsSummaryEntry 1	The collection of probes to which the jnxRpmResultsSummaryEntry refers. NOTE: No entries are created for collection types that are not supported or not configured.
jnxRpmResSumSent	jnxRpmResultsSummaryEntry 2	The number of probes sent within the collection.

Table 320: `jnxRpmResultsSummaryTable` (*continued*)

Object	Object Identifier	Description
<code>jnxRpmResSumReceived</code>	<code>jnxRpmResultsSummaryEntry 3</code>	The number of probes received within the collection.
<code>jnxRpmResSumPercentLost</code>	<code>jnxRpmResultsSummaryEntry 4</code>	The percentage of probes that are lost within the collection.
<code>jnxRpmResSumDate</code>	<code>jnxRpmResultsSummaryEntry 5</code>	The timestamp for the most recent probe within the collection.

`jnxRpmResultsCalculatedTable`

The `jnxRpmResultsCalculatedTable` provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported measurement set within that collection of probes.

The `jnxRpmResultsCalculatedEntry` has the objects listed in [Table 321 on page 755](#).

Table 321: `jnxRpmResultsCalculatedTable`

Object	Object Identifier	Description
<code>jnxRpmResCalcSet</code>	<code>jnxRpmResultsCalculatedEntry 1</code>	The measurement set for the particular <code>jnxRpmResultsCalculatedEntry</code> .
<code>jnxRpmResCalcSamples</code>	<code>jnxRpmResultsCalculatedEntry 2</code>	The number of samples used in the calculations.
<code>jnxRpmResCalcMin</code>	<code>jnxRpmResultsCalculatedEntry 3</code>	The minimum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
<code>jnxRpmResCalcMax</code>	<code>jnxRpmResultsCalculatedEntry 4</code>	The maximum (in microseconds) of all the samples in the collection and the measurement set.
<code>jnxRpmResCalcAverage</code>	<code>jnxRpmResultsCalculatedEntry 5</code>	The average (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
<code>jnxRpmResCalcPkToPk</code>	<code>jnxRpmResultsCalculatedEntry 6</code>	The difference (in microseconds) between the minimum and maximum of all the samples in the collection and the measurement set associated with the entry.
<code>jnxRpmResCalcStdDev</code>	<code>jnxRpmResultsCalculatedEntry 7</code>	The standard deviation (in microseconds) calculated over all the samples in the collection and the measurement set associated with the entry.
<code>jnxRpmResCalcSum</code>	<code>jnxRpmResultsCalculatedEntry 8</code>	The sum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.

jnxRpmHistorySampleTable

The **jnxRpmHistorySampleTable** provides measurements for each sample stored in the history table of RPM probe entries. In addition to the last completed probe, the table also provides data for a configurable number of most recent probes (all the history tables in this MIB provide the same number of entries as the **pingProbeHistoryTable**). However, the table does not maintain entries for:

- Unsuccessful probes
- Invalid measurement types

The **jnxRpmHistorySampleEntry** has the objects listed in [Table 322 on page 756](#).

Table 322: jnxRpmHistorySampleTable

Object	Object Identifier	Description
jnxRpmHistSampleType	jnxRpmHistorySampleEntry 1	The measurement type associated with the entry.
jnxRpmHistSampleValue	jnxRpmHistorySampleEntry 2	The measurement for the entry.
jnxRpmHistSampleTsType	jnxRpmHistorySampleEntry 3	The type of timestamp used to obtain the measurement.

jnxRpmHistorySummaryTable

Similar to the **jnxRpmResultsSummaryTable**, the **jnxRpmHistorySummaryTable** provides you with summary data for each collection of probes within each RPM entry. In addition to summary data for the current probe, the table also provides summary information for a number of the most recent probes. You can configure the number of most recent probes that should be stored in the table.

The **jnxRpmHistorySummaryEntry** has the objects listed in [Table 323 on page 756](#).

Table 323: jnxRpmHistorySummaryTable

Object	Object Identifier	Description
jnxRpmHistSumCollection	jnxRpmHistorySummaryEntry 1	The collection of probes associated with the entry. NOTE: Historical summaries are available only for the current test (currentTest).
jnxRpmHistSumSent	jnxRpmHistorySummaryEntry 2	The number of probes sent within the collection.
jnxRpmHistSumReceived	jnxRpmHistorySummaryEntry 3	The number of probes received within the collection.
jnxRpmHistSumPercentLost	jnxRpmHistorySummaryEntry 4	The percentage of probes lost within the collection.

jnxRpmHistoryCalculatedTable

As with the **jnxRpmResultsCalculatedTable**, the **jnxRpmHistoryCalculatedTable** provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported calculated type within that collection of probes.

In addition to data from the current probe, this table also provides data from a configurable number of the most recent probes.



NOTE: The only collection type that is stored in **jnxRpmHistoryCalculatedTable** is the **currentTest**.

Each **jnxRpmHistoryCalculatedEntry** has the objects listed in [Table 324 on page 757](#).

Table 324: jnxRpmHistoryCalculatedTable

Object	Object Identifier	Description
jnxRpmHistCalcSet	jnxRpmHistoryCalculatedEntry 1	The measurement set for the jnxRpmHistoryCalculatedEntry .
jnxRpmHistCalcSamples	jnxRpmHistoryCalculatedEntry 2	The number of samples used in the calculations for this entry.
jnxRpmHistCalcMin	jnxRpmHistoryCalculatedEntry 3	The minimum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcMax	jnxRpmHistoryCalculatedEntry 4	The maximum (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcAverage	jnxRpmHistoryCalculatedEntry 5	The average (in microseconds) of all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcPkToPk	jnxRpmHistoryCalculatedEntry 6	The difference (in microseconds) between the minimum and the maximum of all the samples in the collection and the measurement set associated with the row.
jnxRpmHistCalcStdDev	jnxRpmHistoryCalculatedEntry 7	The standard deviation (in microseconds) calculated over all the samples in the collection and the measurement set associated with the entry.
jnxRpmHistCalcSum	jnxRpmHistoryCalculatedEntry 8	The sum of all the samples in the collection and the measurement set associated with the entry.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Reverse Path Forwarding MIB

- [Interpreting the Enterprise-Specific Reverse Path Forwarding MIB on page 759](#)

Interpreting the Enterprise-Specific Reverse Path Forwarding MIB

- [Reverse Path Forwarding MIB on page 759](#)
- [jnxRpfStatsTable on page 759](#)

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/junos13.2/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 759](#)

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 325 on page 759](#).

Table 325: 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.

Table 325: jnxRpfStatsEntry (continued)

Object	Object Identifier	Description
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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

RMON Events and Alarms MIB

- [Interpreting the Enterprise-Specific RMON Events and Alarms MIB on page 761](#)

Interpreting the Enterprise-Specific RMON Events and Alarms MIB

- [RMON Events and Alarms MIB on page 761](#)
- [jnxRmonAlarmTable on page 761](#)
- [RMON Event and Alarm Traps on page 763](#)

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/junos13.2/topics/reference/mibs/mib-jnx-rmon.txt.

For more information about RMON alarms and events, see the *Network Management Administration Guide for Routing Devices*.

For information about the enterprise-specific RMON Events and Alarms MIB objects, see the following topics:

- [jnxRmonAlarmTable on page 761](#)
- [RMON Event and Alarm Traps on page 763](#)

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 326 on page 762](#).

Table 326: 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	<p>Represents the reason an internal Get request for the variable monitored by this entry last failed. This object contains the following values:</p> <ul style="list-style-type: none"> • 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.

Table 326: jnxRmonAlarmEntry (*continued*)

Object	Object Identifier	Description
jnxRmonAlarmState	jnxRmonAlarmEntry 5	<p>Represents the current state of this RMON alarm entry. This object contains the following values:</p> <ul style="list-style-type: none"> • unknown (1)—Alarm entry unknown • underCreation (2)—Alarm entry not activated • active (3)—Alarm entry active and within thresholds • startup (4)—Alarm entry still waiting for first value • risingThreshold (5)—Alarm entry has crossed the rising threshold. • fallingThreshold (6)—Alarm entry has crossed the falling threshold • getFailure (7)—Alarm entry internal Get request failed.

RMON Event and Alarm Traps

The following traps send notifications when there is a problem with RMON alarm processing and are listed in [Table 327 on page 763](#).

Table 327: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 66

RSVP MIB

- Interpreting the Enterprise-Specific RSVP MIB on page 765

Interpreting the Enterprise-Specific RSVP MIB

- RSVP MIB on page 765
- `jnxRsvpSessionTable` on page 765

RSVP MIB

The Juniper Networks enterprise-specific RSVP MIB, whose object identifier is **{jnxMibs 30}**, contains information about RSVP-traffic engineering (TE) sessions that correspond to MPLS LSPs on transit routers in the service provider core network.



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

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rsvp.txt.

For information about the enterprise-specific RSVP MIB objects, see the following topic:

- `jnxRsvpSessionTable` on page 765

`jnxRsvpSessionTable`

The `jnxRsvpSessionTable`, whose object identifier is **{jnxRsvpOperation 1}**, contains information about RSVP sessions. Each `jnxRsvpSessionEntry` (object identifier **{jnxRsvpSessionTable 1}**) is identified using a combination of two indexes, `jnxRsvpSessionName` and `jnxRsvpSessionIndex`. The `jnxRsvpSessionName` maps to the LSP name in MPLS entry, and can be used to correlate the `jnxRsvpSessionEntry` with `mplsLspEntry`. Because there can be multiple entries with the same RSVP session name, a secondary index, `jnxRsvpSessionIndex`, is used to uniquely identify each session in combination with the `jnxRsvpSessionName`.

Each `jnxRsvpSessionEntry` contains the objects listed in [Table 328 on page 766](#).

Table 328: jnxRsvpSessionTable

Object	Object ID	Description
jnxRsvpSessionName	jnxRsvpSessionEntry 1	Contains the name of the RSVP session. This is the same as the LSP name in the mplsLspEntry and can contain up to 64 characters. NOTE: The jnxRsvpSessionName object cannot be accessed by running SNMP get and walk commands on the jnxRsvpSessionTable .
jnxRsvpSessionIndex	jnxRsvpSessionEntry 2	Uniquely identifies an RSVP session entry in combination with the jnxRsvpSessionName .
jnxRsvpSessionState	jnxRsvpSessionEntry 3	Shows the operational state of the RSVP session. This object contains one of the following integer values: <ul style="list-style-type: none"> • 1—Up • 2—Down
jnxRsvpSessionFrom	jnxRsvpSessionEntry 4	Contains the source IP address of the RSVP session.
jnxRsvpSessionTo	jnxRsvpSessionEntry 5	Contains the destination IP address of the RSVP session.
jnxRsvpSessionLspId	jnxRsvpSessionEntry 6	Contains the LSP ID of the sender for the RSVP session.
jnxRsvpSessionTunnelId	jnxRsvpSessionEntry 7	Contains the tunnel ID for the RSVP session.
jnxRsvpSessionPathType	jnxRsvpSessionEntry 8	Denotes the type of the path for the RSVP session. This object uses the following integer values to denote the path type: <ul style="list-style-type: none"> • 1—Primary • 2—Secondary • 3—unknown
jnxRsvpSessionRole	jnxRsvpSessionEntry 9	Shows the role of an RSVP session with respect to the start and end points of the session. This object uses the following integer values to represent the role of the RSVP session: <ul style="list-style-type: none"> • 1—Ingress (source) • 2—Transit (intermediate nodes) • 3—Egress (destination)
jnxRsvpSessionDiscontinuityTime	jnxRsvpSessionEntry 10	Shows the value of sysUpTime when either jnxRsvpSessionMplsOctets or jnxRsvpSessionMplsPackets counters experienced discontinuity. This object contains a zero value if no discontinuity occurred since the last initialization of the local management subsystem.

Table 328: jnxRsvpSessionTable (*continued*)

Object	Object ID	Description
jnxRsvpSessionMplsOctets	jnxRsvpSessionEntry 11	Contains the number of MPLS octets that have been forwarded over the RSVP session. Because the MPLS statistics collection occurs at predefined intervals (default of 5 minutes), the value of this object might not reflect real-time statistics. This object is not updated if MPLS statistics collection is not enabled.
jnxRsvpSessionMplsPackets	jnxRsvpSessionEntry 12	Shows the number of MPLS packets that have been forwarded over the RSVP session. Because the MPLS statistics collection occurs at predefined intervals (default of 5 minutes), the value of this object might not reflect real-time statistics. This object is not updated if MPLS statistics collection is not enabled.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Security Interface Extension Objects MIB

- [Interpreting the Enterprise-Specific Security Interface Extension Objects MIB on page 769](#)

Interpreting the Enterprise-Specific Security Interface Extension Objects MIB

- [Security Interface Extension Objects MIB on page 769](#)
- [jnxJsIfMonTable on page 769](#)

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/junos13.2/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 769](#)

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 329 on page 770](#). The table is indexed by **ifIndex**.

Table 329: jnxJsIfMonTable

Object	Object ID	Description
jnxJsIfMonEntry	jnxJsIfMonTable 1	<p>Object related to interface monitoring.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • 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 • jnxJsIfMonSalnactive • 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.

Table 329: 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.

Table 329: 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.
jnxJsIfMonSalnactive	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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 68

Security Screening Objects MIB

- Interpreting the Enterprise-Specific Security Screening Objects MIB on page 773

Interpreting the Enterprise-Specific Security Screening Objects MIB

- Security Screening Objects MIB on page 773
- jnxJsScreenMonTable on page 774
- jnxJsScreenSweepTable on page 785
- jnxJsScreenNotifications on page 786
- jnxJsScreenTrapVars on page 786

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/junos13.2/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 774](#)
- [jnxJsScreenSweepTable on page 785](#)
- [jnxJsScreenNotifications on page 786](#)
- [jnxJsScreenTrapVars on page 786](#)

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 "[jnxJsLoadedCaCertTable](#)" on page 859.

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 330: jnxJsScreenMonTable

Object	Object ID	Description
jnxJsScreenMonEntry	jnxJsScreenMonTable 1	<p>The screen option monitoring statistics entry. Each entry is uniquely identified by the zone name.</p> <p>The data is collected on a per zone basis. There can be multiple interfaces bound to a particular zone. Hence, the statistics are aggregated across the interfaces on a per zone basis.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • jnxJsScreenZoneName • jnxJsScreenNumOfIf • jnxJsScreenMonSynAttk • jnxJsScreenMonTearDrop • jnxJsScreenMonSrcRoute • jnxJsScreenMonPingDeath • jnxJsScreenMonAddrSpoof • jnxJsScreenMonLand • jnxJsScreenMonIcmpFlood • jnxJsScreenMonUdpFlood • jnxJsScreenMonWinNuke • jnxJsScreenMonPortScan • jnxJsScreenMonIpSweep • 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)

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
		<ul style="list-style-type: none"> • jnxJsScreenMonLimitSessDest—Session limit (destination IP-based) • jnxJsScreenMonSynAckAck • jnxJsScreenMonIpFrag • jnxJsScreenSynAttackThresh—Threshold data • jnxJsScreenSynAttackTimeout—Threshold data • jnxJsScreenSynAttackAlmTh—Threshold data • jnxJsScreenSynAttackQueueSize—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	<p>Number of SYN (TCP connection request) attacks.</p> <p>A SYN attack is a common denial of service (DoS) technique characterized by the following pattern:</p> <ul style="list-style-type: none"> • 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 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	<p>Number of teardrop attacks.</p> <p>Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position of the data contained in a fragmented packet relative to the data of the original unfragmented packet. When the sum of the offset and size of one fragmented packet 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.</p> <p>When this option is enabled, the security device detects this discrepancy in a fragmented packet and drops it, and counts the number of packets dropped.</p>

Table 330: jnxJsScreenMonTable (*continued*)

Object	Object ID	Description
jnxJsScreenMonSrcRoute	jnxJsScreenMonEntry 5	<p>Number of either loose source route option packets or strict source route attack packets.</p> <p>IP source route options can be used to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.</p>
jnxJsScreenMonPingDeath	jnxJsScreenMonEntry 6	<p>Number of ping-of-death attack packets.</p> <p>The maximum allowable IP packet size is 65,535 bytes, including the packet header (typically 20 bytes long). An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes.</p> <p>Many ping implementations, however, allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions, such as DoS, crashing, freezing, and rebooting.</p> <p>When the ping-of-death option is enabled, the security device detects and rejects such oversized and irregular packet sizes, even when the attacker hides the total packet size by purposefully fragmenting it.</p>
jnxJsScreenMonAddrSpoof	jnxJsScreenMonEntry 7	<p>Number of address spoofing attack packets.</p> <p>One method to gain access to a restricted network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. The mechanism to detect IP spoofing relies on route table entries.</p> <p>For example, if a packet with source IP address 10.1.1.6 arrives at port 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.</p>
jnxJsScreenMonLand	jnxJsScreenMonEntry 8	<p>Number of land attack packets.</p> <p>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.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenMonIcmpFlood	jnxJsScreenMonEntry 9	<p>Number of ICMP flood attack packets.</p> <p>An ICMP flood typically occurs when ICMP echo requests overload a victim with so many requests that the victim expends all its resources responding to the ICMP echo requests until it can no longer process valid network traffic. With ICMP flood protection enabled and a threshold set, if the threshold is exceeded, the victim invokes the flood attack protection feature.</p> <p>The default threshold value is 1000 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.</p>
jnxJsScreenMonUdpFlood	jnxJsScreenMonEntry 10	<p>Number of UDP flood attack packets.</p> <p>UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. With UDP flood protection enabled, a threshold can be set so that when the threshold is exceeded, the system invokes UDP flood attack protection.</p> <p>The default threshold value is 1000 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.</p>
jnxJsScreenMonWinnuke	jnxJsScreenMonEntry 11	<p>Number of NetBIOS attacks.</p> <p>WinNuke is a DoS attack targeting any computer on the Internet running Microsoft Windows. The attacker sends a TCP segment, usually to NetBIOS port 139 of a host with an established connection with segment's urgent (URG) flag set. This practice introduces a NetBIOS fragment overlap, which causes many machines running Microsoft Windows to crash.</p>
jnxJsScreenMonPortScan	jnxJsScreenMonEntry 12	<p>Number of port scan attempt attack packets.</p> <p>A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval. The purpose of this attack is to scan the available services in the hope that at least one port will respond, thus identifying a service of the target. The security device should internally log the number of different ports scanned from one remote source.</p>
jnxJsScreenMonIpSweep	jnxJsScreenMonEntry 13	<p>Number of address sweep attempt attack packets.</p> <p>An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined interval. The purpose of this attack is to send ICMP packets, typically echo requests, to various hosts in the hope that at least one replies, thus uncovering an address of the target. The security device internally logs the number of ICMP packets to different addresses from one remote source.</p>

Table 330: jnxJsScreenMonTable (*continued*)

Object	Object ID	Description
jnxJsScreenMonSynFrag	jnxJsScreenMonEntry 14	<p>Number of SYN fragments.</p> <p>IP encapsulates a TCP SYN segment in the IP packet that initiates a TCP connection. The purpose is to initiate a connection and to invoke a SYN/ACK segment response. The SYN segment typically does not contain any data since the IP packet is small and there is no legitimate reason for it to be fragmented. A fragmented SYN packet is anomalous and is suspicious. To be cautious, it might be helpful to block such fragments from entering the protected network.</p> <p>When the SYN fragmentation check is enabled, the security device detects and drops the packets when the IP header indicates that the packet has been fragmented while the SYN flag is set in the TCP header.</p>
jnxJsScreenMonTcpNoFlag	jnxJsScreenMonEntry 15	<p>Number of TCP packets with no flag set.</p> <p>A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Operating systems respond to such anomalies in different ways. The response, or even lack of response, from the targeted device can provide a clue as to the target's OS type.</p> <p>When this option is enabled, if the security device discovers such a header with a missing or malformed flags field, it drops the packet.</p>
jnxJsScreenMonIpUnknownProt	jnxJsScreenMonEntry 16	<p>Number of of unknown protocol IP packets.</p> <p>According to RFC 1700, some protocol types in an IP header are reserved and unassigned at this time. Precisely because these protocols are undefined, there is no way to know in advance whether a particular unknown protocol is benign or malicious. Unless your network makes use of a nonstandard protocol with a reserved or unassigned protocol number, a cautious stance is to block such unknown elements from entering your protected network.</p> <p>When the Unknown Protocol Protection SCREEN option is enabled, the security device drops packets when the protocol field contains a protocol ID number of 137 or greater.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenMonIpOptBad	jnxJsScreenMonEntry 17	<p>Number of IP bad option packets.</p> <p>The IP protocol specifies a set of eight options that provide special routing controls, diagnostic tools, and security. These eight options can be used for malicious objectives.</p> <p>Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. The incorrect formatting is anomalous and potentially harmful to the intended recipient.</p> <p>When the Bad IP Option Protection SCREEN option is enabled, the security device detects and blocks packets when any IP option in the IP packet header is incorrectly formatted.</p>
jnxJsScreenMonIpOptRecRt	jnxJsScreenMonEntry 18	<p>Number of IP record option packets.</p> <p>The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. The record option is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a record IP option is received, the security device flags it as a network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIpOptTimestamp	jnxJsScreenMonEntry 19	<p>Number of IP timestamp option packets.</p> <p>The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Timestamp is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a timestamp IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIpOptSecurity	jnxJsScreenMonEntry 20	<p>Number of IP security option packets.</p> <p>The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Security is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a security IP option is received, the security device flags this as a network reconnaissance attack and records the event for the ingress interface.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenMonIpOptLSR	jnxJsScreenMonEntry 21	<p>Number of strict source route packets.</p> <p>Attackers can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. The security device should be able to either block any packets with loose or strict source route options set or detect such packets and then record the event for the ingress interface.</p>
jnxJsScreenMonIpOptStream	jnxJsScreenMonEntry 23	<p>Number of IP stream option packets.</p> <p>The IP standard RFC 791 specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header. When they do appear, they are frequently being put to some nefarious use. Stream is one of these options that an attacker can use for reconnaissance or for some unknown but suspicious purpose.</p> <p>When a security IP option is received, the security device flags it as a network reconnaissance attack and records the event for the ingress interface.</p>
jnxJsScreenMonIcmpFrag	jnxJsScreenMonEntry 24	<p>Number of ICMP fragment packets.</p> <p>ICMP provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is wrong. With the ICMP Fragment Protection SCREEN option enabled, the security device should be able to block any ICMP packet with the More Fragments flag set or with an offset value indicated in the offset field.</p>
jnxJsScreenMonIcmpLarge	jnxJsScreenMonEntry 25	<p>Number of large ICMP packets.</p> <p>Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented.</p> <p>If an ICMP packet is unusually large, something is wrong. For example, the Loki program uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a Loki agent. It might also indicate some other kind of malicious activity.</p> <p>When the the Large Size ICMP Packet Protection SCREEN option is enabled, the security device drops ICMP packets with a length greater than 1024 bytes.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenMonTcpSynFin	jnxJsScreenMonEntry 26	<p>Number of dropped TCP packets because SYN and FIN are both set.</p> <p>Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS.</p> <p>When the blocking of TCP packets with both SYN and FIN is enabled, the security device drops the packet when it discovers such a header.</p>
jnxJsScreenMonTcpFinNoAck	jnxJsScreenMonEntry 27	<p>Number of TCP packets with FIN set, but without the ACK bit set.</p> <p>A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response and thereby discover an active host or an active port on a host. The use of TCP segments with the FIN flag set might evade detection and thereby help attackers succeed in their reconnaissance efforts.</p>
jnxJsScreenMonLimitSessSrc	jnxJsScreenMonEntry 28	<p>Number of session connections for a source IP address that exceeds the specified limit.</p> <p>Because all the virus-generated traffic originates from the same IP address (generally from an infected server), a source-based session limit ensures that the firewall can curb such excessive amounts of traffic. This amount is based on a threshold value of the number of concurrent sessions required to fill up the session table of the particular firewall.</p> <p>The default maximum for a source-based session limit is 128 concurrent sessions, which can be adjusted accordingly.</p>
jnxJsScreenMonLimitSessDest	jnxJsScreenMonEntry 29	<p>Number of session connections for the destination source IP address that exceeds the specified limit.</p> <p>The user can limit the number of concurrent sessions to the same destination IP address. An attacker can launch a distributed denial of service (DDoS) attack using "zombie agents." Setting a destination-based session limit can ensure that the security device allows only an acceptable number of concurrent connection requests, no matter what the source, to reach any one host.</p> <p>The default maximum for the destination-based session limit is 128 concurrent sessions.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenMonSynAckAck	jnxJsScreenMonEntry 30	<p>Number of SYN ACK ACK attacks.</p> <p>When an authentication user initiates a Telnet or FTP connection, the user sends a SYN segment to the Telnet or FTP server. The security device intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At that point, the initial three-way handshake is complete. The security device sends a login prompt to the user. When a malicious user does not log in, but instead continues initiating SYN-ACK-ACK sessions, the firewall session table can fill up to the point at which the security device begins rejecting legitimate connection requests.</p> <p>When the SYN-ACK-ACK proxy protection option is enabled, after the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, the security device rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address.</p>
jnxJsScreenMonIpFrag	jnxJsScreenMonEntry 31	<p>Number of block IP fragment packets.</p> <p>As a packets travels, it is sometimes necessary to break the packet into smaller fragments based 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.</p> <p>When the block IP fragmentation flag is enabled, the security device blocks all IP packet fragments that it receives at interfaces bound to that zone.</p>
Threshold Values		
jnxJsScreenSynAttackThresh	jnxJsScreenMonEntry 32	<p>SYN attack threshold value.</p> <p>The number of SYN segments to the same destination address and port number per second required to activate the SYN proxying mechanism. Setting the appropriate threshold value requires a thorough knowledge of the normal traffic patterns at the site.</p> <p>For example, if the security device normally gets 2000 SYN segments per second, the threshold value should be set at 3000 segments per second.</p>
jnxJsScreenSynAttackTimeout	jnxJsScreenMonEntry 33	<p>SYN attack timeout value.</p> <p>The maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds.</p>

Table 330: jnxJsScreenMonTable (*continued*)

Object	Object ID	Description
jnxJsScreenSynAttackAlmTh	jnxJsScreenMonEntry 34	<p>SYN attack alarm threshold value.</p> <p>The SYN attack alarm threshold causes an alarm to be generated when the number of proxied, half-completed TCP connection requests per second to the same destination address and port number exceeds its value.</p>
jnxJsScreenSynAttackQueueSize	jnxJsScreenMonEntry 35	<p>SYN attack queue size.</p> <p>The number of proxied connection requests held in the proxied connection queue before the security device starts rejecting new connection requests.</p>
NOTE: The jnxJsScreenSynAttackAgeTime object is obsolete in this release.		
jnxJsScreenSynAttackAgeTime	jnxJsScreenMonEntry 36	SYN flood age time.
jnxJsScreenIcmpFloodThresh	jnxJsScreenMonEntry 37	<p>ICMP attack alarm threshold value.</p> <p>The security device can impose a limit on the number of SYN segments permitted to pass through the firewall per second. The default attack threshold value is 1000. The valid threshold range is 1 through 100,000. When the threshold value is exceeded, an alarm is triggered.</p>
jnxJsScreenUdpFloodThresh	jnxJsScreenMonEntry 38	<p>UDP attack alarm threshold value.</p> <p>UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections.</p> <p>The default threshold value is 1000 pps.</p>
jnxJsScreenPortScanThresh	jnxJsScreenMonEntry 39	<p>Port scan threshold value.</p> <p>The port scan threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1,000,000.</p> <p>By using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), the security device flags this occurrence as a port scan attack and rejects all further packets from the remote source for the remainder of the specified timeout period. The security device detects and drops the tenth packet that meets the port scan attack criterion.</p>

Table 330: jnxJsScreenMonTable (continued)

Object	Object ID	Description
jnxJsScreenIpSweepThresh	jnxJsScreenMonEntry 40	<p>IP sweep threshold interval.</p> <p>The IP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000 through 1,000,000.</p> <p>By using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this occurrence as an address sweep attack and rejects all further ICMP echo requests from that host for the remainder of the specified threshold time period. The security device detects and drops the tenth packet that meets the address sweep attack criterion.</p>
jnxJsScreenSynAckAckThres	jnxJsScreenMonEntry 41	SYN-ACK-ACK alarm threshold value.

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 331 on page 785](#).

Table 331: jnxJsScreenSweepTable

Object	Object ID	Description
jnxJsScreenSweepEntry	jnxJsScreenSweepTable 1	<p>Maintains the TCP/UDP sweep thresholds and counters.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • jnxJsScreenTcpSweepThresh • jnxJsScreenUdpSweepThresh • jnxJsScreenMonTcpSweep • jnxJsScreenMonUdpSweep
jnxJsScreenTcpSweepThresh	jnxJsScreenSweepEntry 1	<p>The TCP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000-1,000,000.</p> <p>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.</p>

Table 331: jnxJsScreenSweepTable (*continued*)

Object	Object ID	Description
jnxJsScreenUdpSweepThresh	jnxJsScreenSweepEntry 2	<p>The UDP sweep threshold interval is in microseconds. The default threshold value is 5000. The valid threshold range is 1000-1,000,000.</p> <p>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.</p>
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.

jnxJsScreenNotifications

The **jnxJsScreenNotifications** object, whose object ID is **{jnxJsScreenNotifications 0}**, specifies the screen notifications listed in [Table 332 on page 786](#).

Table 332: jnxJsScreenNotifications

Object	Object ID	Description
jnxJsScreenAttack	jnxJsScreenNotifications 1	<p>Specifies that the number of bytes per minute has exceeded the specified threshold.</p> <ul style="list-style-type: none"> • 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	<p>Specifies the status of the screening option.</p> <ul style="list-style-type: none"> • jnxJsScreenCfgStatus — Specifies if the screening option is enabled or disabled.

jnxJsScreenTrapVars

The **jnxJsScreenTrapVars** object identifies the screen trap variables listed in [Table 333 on page 787](#).

Table 333: jnxJsScreenTrapVars

Object	Object ID	Description
jnxJsScreenAttackType	jnxJsScreenTrapVars 1	Returns one of the following integer values to identify the type of attack: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 1—disabled • 2—enabled

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Services PIC MIB

- Interpreting the Enterprise-Specific Services PIC MIB on page 789

Interpreting the Enterprise-Specific Services PIC MIB

- Services PIC MIB on page 789
- jnxSpSvcSetTable on page 790
- jnxSpSvcSetSvcTypeTable on page 791
- jnxSpSvcSetIfTable on page 792
- Service Traps on page 793
- Redundant Interfaces on page 793

Services PIC MIB

The Adaptive Services (AS) PIC allows you to provide multiple services on a single PIC by configuring a set of services and applications. The AS PIC offers a special range of services you configure in one or more service sets: stateful firewalls, Network Address Translation (NAT), and intrusion detection system (IDS).

The Juniper Networks enterprise-specific Services PIC MIB, whose object identifier is **{jnxMibs 32}**, sends the current operational status for each AS PIC. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sp.txt.



NOTE: The Services PIC MIB is not supported on J Series Services Routers unless the appropriate services license is enabled.

For information about the enterprise-specific Services PIC MIB objects, see the following topics:

- jnxSpSvcSetTable on page 790
- jnxSpSvcSetSvcTypeTable on page 791
- jnxSpSvcSetIfTable on page 792

- [Service Traps on page 793](#)
- [Redundant Interfaces on page 793](#)

jnxSpSvcSetTable

The **jnxSpSvcSetTable**, whose object identifier is **{jnxSPSvcSet 1}**, provides information about each service set on each Adaptive Services (AS) PIC on the router.

The **jnxSpSvcSetEntry**, whose object identifier is **{jnxSpSvcSetTable 1}**, has 11 objects, which are listed in [Table 334 on page 790](#). Each entry provides information about a single service set. The service set is identified by the name of the service set. The AS PIC on which the service set is configured is identified by **JnxSpSvcSetIfName**.

Table 334: jnxSpSvcSetTable

Object	Object Identifier	Description
jnxSpSvcSetName	jnxSpSvcSetEntry 1	A text name for the service set.
jnxSpSvcSetSvcType	jnxSpSvcSetEntry 2	The name of the service type associated with the service set.
jnxSpSvcSetTypeIndex	jnxSpSvcSetEntry 3	An integer used to identify the service type for the service set.
jnxSpSvcSetIfName	jnxSpSvcSetEntry 4	The name of the interface identifying the AS PIC. If more than one interface is associated with the AS PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetIfIndex	jnxSpSvcSetEntry 5	An index number associated with the interface name.
jnxSpSvcSetMemoryUsage	jnxSpSvcSetEntry 6	Amount of memory used by the service set, in bytes.
jnxSpSvcSetCpuUtil	jnxSpSvcSetEntry 7	Amount of CPU processing used by the service set, expressed as a percentage of total CPU usage. J Series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.
jnxSpSvcSetSvcStyle	jnxSpSvcSetEntry 8	Type of service for the service set. Service types include: <ul style="list-style-type: none"> • Unknown—The service type is not known. • Interface-service—The service is interface based. • Next-hop-service—The service is next-hop based.

Table 334: jnxSpSvcSetTable (*continued*)

Object	Object Identifier	Description
jnxSpSvcSetMemLimitPktDrops	jnxSpSvcSetEntry 9	Number of packets dropped because the service set exceeded its memory limits (operating in the Red zone).
jnxSpSvcSetCpuLimitPktDrops	jnxSpSvcSetEntry 10	Number of packets dropped because the service set exceeded the average CPU limits (when total CPU usage exceeds 85%).
jnxSpSvcSetFlowLimitPktDrops	jnxSpSvcSetEntry 11	Number of packets dropped because the service set exceeded the flow limit.

jnxSpSvcSetSvcTypeTable

The **jnxSpSvcSetSvcTypeTable**, whose object identifier is **{jnxSPSvcSet 2}**, provides information about each service on each Adaptive Services (AS) PIC on the router. The stateful firewall, NAT, or IDS service sets are categorized as one **SvcType** (SFW/NAT/IDS).

The **jnxSpSvcSetSvcTypeEntry**, whose object identifier is **{jnxSpSvcSetSvcTypeTable 1}**, has seven objects, which are listed in [Table 335 on page 791](#). Each entry provides information about a single service on each AS PIC. Each AS PIC is identified by its corresponding index number, while each service is identified by **jnxSpSvcSetSvcTypeIndex**. The service type associated with this index is provided by **jnxSpSvcSetSvcTypeName**.

Table 335: jnxSpSvcSetSvcTypeTable

Object	Object Identifier	Description
jnxSpSvcSetSvcTypeIndex	jnxSpSvcSetSvcTypeEntry 1	An integer used to identify the service type.
jnxSpSvcSetSvcTypeIfName	jnxSpSvcSetSvcTypeEntry 2	The name of the interface identifying the AS PIC. If more than one interface is associated with the AS PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetSvcTypeName	jnxSpSvcSetSvcTypeEntry 3	The name of the service type.
jnxSpSvcSetSvcTypeSvcSets	jnxSpSvcSetSvcTypeEntry 4	Number of service sets configured on the AS PIC that use this service type.
jnxSpSvcSetSvcTypeMemoryUsage	jnxSpSvcSetSvcTypeEntry 5	Amount of memory used by this service type, expressed in bytes.
jnxSpSvcSetSvcTypePctMemoryUsage	jnxSpSvcSetSvcTypeEntry 6	Amount of memory used by this service type, expressed as a percentage of total memory.

Table 335: jnxSpSvcSetSvcTypeTable (*continued*)

Object	Object Identifier	Description
jnxSpSvcSetSvcTypeCpuUtil	jnxSpSvcSetSvcTypeEntry 7	Amount of CPU processing used by the service set, expressed as a percentage of total CPU usage. J Series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.

jnxSpSvcSetIfTable

The **jnxSpSvcSetIfTable**, whose object identifier is **{jnxSPSvcSet 3}**, provides service set information for each Adaptive Services (AS) PIC on the router.

The **jnxSpSvcSetIfEntry**, whose object identifier is **{jnxSpSvcSetIfTable 1}**, has eight objects, which are listed in [Table 336 on page 792](#). Each entry provides service set information about a single AS PIC. Each AS PIC is identified by its corresponding index number.

Table 336: jnxSpSvcSetIfTable

Object	Object Identifier	Description
jnxSpSvcSetIfTableName	jnxSpSvcSetIfEntry 1	The name of the interface used to identify the AS PIC. If more than one interface is associated with the AS PIC, the name associated with the lower layer interface is used.
jnxSpSvcSetIfSvcSets	jnxSpSvcSetIfEntry 2	The number of service sets configured on the AS PIC.
jnxSpSvcSetIfMemoryUsage	jnxSpSvcSetIfEntry 3	Amount of memory used by the AS PIC, expressed in bytes.
jnxSpSvcSetIfPctMemoryUsage	jnxSpSvcSetIfEntry 4	Amount of memory used by the AS PIC, expressed as a percentage of total memory.
jnxSpSvcSetIfPolMemoryUsage	jnxSpSvcSetIfEntry 5	Amount of policy memory used by the AS PIC, expressed in bytes.
jnxSpSvcSetIfPctPolMemoryUsage	jnxSpSvcSetIfEntry 6	Amount of policy memory used by the AS PIC, expressed as a percentage of the total.
jnxSpSvcSetIfMemoryZone	jnxSpSvcSetIfEntry 7	The memory usage zone currently occupied by the AS PIC. The definitions of each zone are: <ul style="list-style-type: none"> • Green—All new flows are allowed. • Yellow—Unused memory is reclaimed. All new flows are allowed. • Orange—New flows are allowed only for service sets that use less than their equal share of memory. • Red—No new flows are allowed.

Table 336: jnxSpSvcSetIfTable (*continued*)

Object	Object Identifier	Description
jnxSpSvcSetIfCpuUtil	jnxSpSvcSetIfEntry 8	Amount of CPU processing used by the AS PIC, expressed as a percentage of total CPU usage. J Series Services Routers do not have a dedicated CPU for services. CPU usage on these routers appears as 0.

Service Traps

The enterprise-specific Services PIC MIB provides traps for monitoring Adaptive Services (AS) PICs. [Table 337 on page 793](#) lists the supported traps.

Table 337: Supported Traps for Services PIC MIB

Object	Object Identifier	Description
jnxSpSvcSetZoneEntered	jnxSPNotificationPrefix 1	Indicates that an AS PIC has entered a more severe memory usage zone from a less severe memory usage zone. The zone entered is identified by JnxSpSvcSetIfMemoryZone .
jnxSpSvcSetZoneExited	jnxSPNotificationPrefix 2	Indicates that an AS PIC has exited a more severe memory usage zone to a less severe memory usage zone. The zone entered is identified by JnxSpSvcSetIfMemoryZone .
jnxSpSvcSetCpuExceeded	jnxSPNotificationPrefix 3	Indicates that an AS PIC has over 85% CPU usage. This trap is not supported on J Series Services Routers.
jnxSpSvcSetCpuOk	jnxSPNotificationPrefix 4	Indicates that an AS PIC has returned to less than 85% CPU usage. This trap is not supported on J Series Services Routers.

Redundant Interfaces

On M Series routers and T Series routers, redundant adaptive services interfaces (**rsp**) appear in the **jnxSpSvcSetIfTable** just like any other adaptive services interface (**sp**). With the exception of the index, information presented for an **rsp** interface is similar to the underlying **sp** interface. In the **jnxSpSvcSetTable**, only the underlying **sp** interface is shown because the Adaptive Services (AS) PIC does not track the overlying **rsp** interface,

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 70

SNMP IDP MIB

- [Interpreting the Enterprise-Specific SNMP IDP MIB on page 795](#)

Interpreting the Enterprise-Specific SNMP IDP MIB

- [SNMP IDP MIB on page 795](#)
- [jnxJsldpObjects on page 795](#)
- [jnxJsldpAttackTable on page 796](#)
- [IDP Signature Update and Attack Traps on page 797](#)

SNMP IDP MIB

The Juniper Networks enterprise-specific SNMP IDP MIB, **jnxJsldpMIB**, whose object ID is **{jnxJsldpRoot 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/junos13.2/topics/reference/mibs/mib-jnx-js-idp.txt.

For information about the enterprise-specific SNMP IDP MIB objects, see the following topics:

- [jnxJsldpObjects on page 795](#)
- [jnxJsldpAttackTable on page 796](#)
- [IDP Signature Update and Attack Traps on page 797](#)

jnxJsldpObjects

You use **jnxJsldpObjects**, whose object ID is **{jnxJsldpMIB 1}**, to display the IDP query-related statistics listed in [Table 338 on page 796](#).

Table 338: jnxJsIdpObjects

Object	Object ID	Description
jnxJsIdpDataPlaneMemoryUsage	jnxJsIdpObjects 1	The percentage of memory used by IDP in the data plane.
jnxJsIdpSessionsUsage	jnxJsIdpObjects 2	The percentage of currently allocated sessions by IDP.
jnxJsIdpSessionsMaximum	jnxJsIdpObjects 3	The maximum number of sessions IDP can support.
jnxJsIdpPoliciesSupported	jnxJsIdpObjects 4	The number of policies that can be simultaneously loaded on the IDP device.
jnxJsIdpPoliciesLoaded	jnxJsIdpObjects 5	The number of policies that are currently loaded on the IDP device.
jnxJsIdpActivePolicyName	jnxJsIdpObjects 6	The name of the policy that is currently active on the device.
jnxJsIdpRunningDetectorVersion	jnxJsIdpObjects 8	The version of the detector that is currently loaded.
jnxJsIdpSecurityPackageVersion	jnxJsIdpObjects 9	The version of the security that is package currently loaded.
jnxJsIdpLastSignatureUpdateTime	jnxJsIdpObjects 10	The value of sysUpTime when the signature was last updated successfully. Zero if unknown.
jnxJsIdpSignatureUpdateStatus	jnxJsIdpObjects 11	The status of the signature update: success or failure.

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 339 on page 796](#).

Table 339: jnxJsIdpAttackTable

Object	Object ID	Description
jnxJsIdpAttackEntry	jnxJsIdpAttackTable 1	The attributes of an attack Sequence of parameters: <ul style="list-style-type: none"> • jnxJsIdpAttackIndex • jnxJsIdpAttackName • jnxJsIdpAttackHits
jnxJsIdpAttackIndex	jnxJsIdpAttackEntry 1	The index for the attack table.

Table 339: jnxJsIdpAttackTable (*continued*)

Object	Object ID	Description
jnxJsIdpAttackName	jnxJsIdpAttackEntry 2	The name of an attack identified by IDP.
jnxJsIdpAttackHits	jnxJsIdpAttackEntry 3	The number of times an attack has been launched.

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 340 on page 797](#) for information about the signature update and attack traps.

Table 340: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

SONET/SDH Interface Management MIB

- Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB on page 799

Interpreting the Enterprise-Specific SONET/SDH Interface Management MIB

- SONET/SDH Interface Management MIB on page 799
- `jnxSonetAlarmsTable` on page 799

SONET/SDH Interface Management MIB

The Juniper Networks enterprise-specific SONET/SDH Interface Management MIB sends the current alarm state for each SONET/SDH interface. When the alarm state changes on an interface, the MIB updates its alarm status. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sonet.txt.

For information about the enterprise-specific SONET/SDH Interface Management MIB objects, see the following topic:

- `jnxSonetAlarmsTable` on page 799

`jnxSonetAlarmsTable`

The `jnxSonetAlarmsTable`, whose object identifier is `{jnxSonetAlarm 1}`, provides information about alarm status on SONET/SDH physical interfaces.

The `jnxSonetAlarmEntry`, whose object identifier is `{jnxSonetAlarmTable 1}`, has five objects, which are listed in [Table 341 on page 799](#).

Table 341: `jnxSonetAlarmTable`

Object	Object Identifier	Description
<code>jnxSonetCurrentAlarms</code>	<code>jnxSonetAlarmEntry 1</code>	Identifies all the active SONET/SDH alarms on this interface.
<code>jnxSonetLastAlarmId</code>	<code>jnxSonetAlarmEntry 2</code>	Identifies the SONET/SDH alarm that most recently was set or cleared.

Table 341: jnxSonetAlarmTable (continued)

Object	Object Identifier	Description
jnxSonetLastAlarmTime	jnxSonetAlarmEntry 3	The value of sysUpTime when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmDate	jnxSonetAlarmEntry 4	The system date and time when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmEvent	jnxSonetAlarmEntry 5	Indicates whether the last alarm event set a new alarm or cleared an existing alarm.

Table 342 on page 800 provides an example of **jnxSonetAlarmInterface** objects of an M20 router.

Table 342: jnxSonetAlarmInterface Objects in the jnxSonetAlarmTable of an M20 Router

Alarm Interface	CurrentAlarms	Last Alarm ID	Last Alarm Time (System Up Time)	Last Alarm Date and Time	Last Alarm Event
14	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.15	2002-10-15, 10:21:14.0,-7:0	set(2)
15	sonetLosAlarm(3)	sonetLosAlarm(3)	8 days, 4:09:46.22	2002-10-23,14:29:23.0,-7:0	set(2)
16	sonetLolAlarm(0) sonetLosAlarm(3)	sonetBerrSdAlarm(8)	8 days, 4:09:46.21	2002-10-23,14:29:23.0,-7:0	cleared(3)
17	sonetLofAlarm(2)	sonetLaisAlarm(5)	8 days, 4:09:47.21	2002-10-23,14:29:24.0,-7:0	cleared(3)
18	—	sonetLosAlarm(3)	7 days, 4:31:27.53	2002-10-22,14:51:4.0,-7:0	cleared(3)
19	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.16	2002-10-15,10:21:14.0,-7:0	set(2)
20	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.17	2002-10-15,10:21:14.0,-7:0	set(2)
21	—	sonetLofAlarm(2)	7 days, 11:15:00.15	2002-10-22,21:34:37.0,-7:0	cleared(3)
22	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLolAlarm(0)	7 days, 6:33:32.02	2002-10-22,16:53:8.0,-7:0	set(2)
23	—	sonetLosAlarm(3)	7 days, 6:33:45.02	2002-10-22,16:53:21.0,-7:0	cleared(3)
24	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.07	2002-10-15,10:21:14.0,-7:0	set(2)

Table 342: jnxSonetAlarmInterface Objects in the jnxSonetAlarmTable of an M20 Router (continued)

Alarm Interface	CurrentAlarms	Last Alarm ID	Last Alarm Time (System Up Time)	Last Alarm Date and Time	Last Alarm Event
25	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.08	2002-10-15,10:21:14.0,-7:0	set(2)
26	—	—	0:00:00.00	0-0-0,0:0:0.0,	none(1)
27	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
28	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
29	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 72

SONET APS MIB

- Interpreting the Enterprise-Specific SONET APS MIB on page 803

Interpreting the Enterprise-Specific SONET APS MIB

- SONET APS MIB on page 803
- apsConfigTable on page 804
- apsStatusTable on page 805
- apsChanConfigTable on page 808
- apsChanStatusTable on page 809

SONET APS MIB

The Juniper Networks enterprise-specific SONET Automatic Protection Switching (APS) MIB monitors any SONET interface that participates in APS. APS is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The Junos OS implementation of APS allows you to protect against circuit failures between an ADM and one or more routers, and between multiple interfaces in the same router. When a circuit or router fails, a backup immediately takes over. For more information about APS, see the *Junos OS Network Interfaces Library for Routing Devices*.



NOTE: The Junos OS supports only read access, 1+1 architecture, bidirectional, revertive, and nonrevertive mode.

For a downloadable version of this MIB, see
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-sonetaps.txt.

For information about the enterprise-specific SONET APS MIB objects, see the following topics:

- apsConfigTable on page 804
- apsStatusTable on page 805
- apsChanConfigTable on page 808
- apsChanStatusTable on page 809

apsConfigTable

apsConfigTable lists the APS groups that are configured on the system.

apsConfigEntry objects have read access only and are listed in [Table 343 on page 804](#).

Table 343: apsConfigTable

Object	Object Identifier	Description
apsConfigName	apsConfigEntry 1	<p>A text name for the APS group.</p> <p>An entry cannot exist in the active state unless all objects in the entry have an appropriate value. Also, all associated apsChanConfigEntry rows must represent a set of consecutive channel numbers beginning with 0 or 1, depending on the selected architecture.</p>
apsConfigRowStatus	apsConfigEntry 2	The status of a APS group entry.
apsConfigMode	apsConfigEntry 3	The architecture of the APS group. Junos OS supports only the 1+1 architecture.
apsConfigRevert	apsConfigEntry 4	<p>The revertive mode of the APS group.</p> <ul style="list-style-type: none"> Revertive mode—When the condition that caused a switch to the protection line has been cleared, the signal is switched back to the working line. Switching can optionally be revertive with 1+1 architecture. Nonrevertive mode—Traffic remains on the protection line until another switch request is received.
apsConfigDirection	apsConfigEntry 5	The directional mode of the APS group. Junos OS supports only bidirectional mode. Bidirectional mode provides protection in both directions.
apsConfigExtraTraffic	apsConfigEntry 6	This object always returns the value disabled.
apsConfigSdBerThreshold	apsConfigEntry 7	The signal degrade bit error rate (BER). The negative value of this number is used as the exponent of 10 for computing the threshold value for the BER. For example, a value of 5 indicates a BER threshold of 10^{-5} .
apsConfigSfBerThreshold	apsConfigEntry 8	The signal failure bit error rate (BER). The negative value of this number is used as the exponent of 10 for computing the threshold value for the BER. For example, a value of 5 indicates a BER threshold of 10^{-5} .

Table 343: `apsConfigTable` (*continued*)

Object	Object Identifier	Description
<code>apsConfigWaitToRestore</code>	<code>apsConfigEntry 9</code>	<p>The wait to restore period, in seconds. After a condition that necessitated an automatic switch is cleared, the wait to restore period must elapse before reverting. This avoids rapid switch oscillations.</p> <p>GR-253-CORE specifies a range of 5 to 12 minutes. G.783 defines a range of 5 to 12 minutes in section 5.4.1.1.3, but also allows a shorter period in Table 2-1, WaitToRestore value (MI_WTRtime: 0..(5)..12 minutes).</p>
<code>apsConfigCreationTime</code>	<code>apsConfigEntry 10</code>	The value of sysUpTime at the time the row was created.
<code>apsConfigStorageType</code>	<code>apsConfigEntry 11</code>	The storage type for this conceptual row. For information about conceptual rows, see RFC 2579, <i>Textual Conventions for SMIv2</i> .

apsStatusTable

`apsStatusTable` provides status information about configured APS groups.

`apsStatusEntry` objects have read access only and are listed in [Table 344 on page 805](#).

Table 344: `apsStatusTable`

Object	Object Identifier	Description
<code>apsStatusK1K2Rcv</code>	<code>apsStatusEntry 1</code>	The current value of the K1 and K2 bytes received on the protection channel.
<code>apsStatusK1K2Trans</code>	<code>apsStatusEntry 2</code>	The current value of the K1 and K2 bytes transmitted on the protection channel.

Table 344: `apsStatusTable` (*continued*)

Object	Object Identifier	Description
<code>apsStatusCurrent</code>	<code>apsStatusEntry 3</code>	<p>The current status of the APS group. This object has the following values:</p> <ul style="list-style-type: none"> • modeMismatch—Modes other than 1+1 unidirectional monitor protection line K2 bit 5, which indicates the architecture, and K2 bits 6 through 8, which indicate whether the mode is unidirectional or bidirectional. A conflict between the current local mode and the received K2 mode information constitutes a mode mismatch. Junos OS supports only bidirectional mode. • channelMismatch—A mismatch between the transmitted K1 channel and the received K2 channel has been detected. • psbf—A protection switch byte failure (PSBF) is in effect. This condition occurs when either an inconsistent APS byte or an invalid code is detected. An inconsistent APS byte occurs when no 3 consecutive K1 bytes of the last 12 successive frames are identical, starting with the last frame containing a previously consistent byte. An invalid code occurs when the incoming K1 byte contains an unused code or a code irrelevant for the specific switching operation (for example, reverse request while no switching request is outstanding) in three consecutive frames. An invalid code also occurs when the incoming K1 byte contains an invalid channel number in three consecutive frames. • feplf—Modes other than 1+1 unidirectional monitor the K1 byte for far-end protection-line failures. A far-end protection-line defect is declared based on receiving a signal failure (SF) on the protection line. • extraTraffic—Indicates whether extra traffic is currently being accepted on the protection line. • extraTraffic—Indicates whether extra traffic is currently being accepted on the protection line.
<code>apsStatusModeMismatches</code>	<code>apsStatusEntry 4</code>	<p>Counts mode mismatch conditions. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of <code>apsStatusDiscontinuityTime</code>.</p>

Table 344: apsStatusTable (*continued*)

Object	Object Identifier	Description
apsStatusChannelMis-matches	apsStatusEntry 5	Counts channel mismatch conditions. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime .
apsStatusPSBFs	apsStatusEntry 6	<p>Counts protection switch byte failure conditions. This condition occurs when either an inconsistent APS byte or an invalid code is detected.</p> <p>An inconsistent APS byte occurs when no 3 consecutive K1 bytes of the last 12 successive frames are identical, starting with the last frame containing a previously consistent byte.</p> <p>An invalid code occurs when the incoming K1 byte contains an unused code or a code irrelevant for the specific switching operation (for example, reverse request while no switching request is outstanding) in three consecutive frames. An invalid code also occurs when the incoming K1 byte contains an invalid channel number in three consecutive frames.</p> <p>Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime.</p>
apsStatusFEPLFs	apsStatusEntry 7	Counts far-end protection-line failure conditions. This condition is declared based on receiving a signal failure (SF) on the protection line in the K1 byte. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsStatusDiscontinuityTime .
apsStatusSwitchedChannel	apsStatusEntry 8	This field is set to the number of the channel that is currently switched to protection. The value 0 indicates that no channel is switched to protection. The values 1 through 14 indicate that the working channel is switched to protection.

Table 344: `apsStatusTable` (*continued*)

Object	Object Identifier	Description
<code>apsStatusDiscontinuity-Time</code>	<code>apsStatusEntry 9</code>	The value of <code>sysUpTime</code> when the last one or more of this APS group's counters experienced a discontinuity. The relevant counters are the specific instances associated with this APS group of any <code>Counter32</code> object contained in <code>apsStatusTable</code> . If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value.

`apsChanConfigTable`

`apsChanConfigTable` lists the APS channels that have been configured in APS groups.

`apsChanConfigEntry` objects have read access only and are listed in [Table 345 on page 808](#).

Table 345: `apsChanConfigEntry`

Object	Object Identifier	Description
<code>apsChanConfigGroupName</code>	<code>apsChanConfigEntry 1</code>	A text name for the APS group in which this channel is included.
<code>apsChanConfigNumber</code>	<code>apsChanConfigEntry 2</code>	A unique channel number within an APS group. The value 0 indicates the null channel. The values 1 through 14 define a working channel.
<code>apsChanConfigRowStatus</code>	<code>apsChanConfigEntry 3</code>	<p>The status of this APS channel entry. An entry cannot exist in the active state unless all objects in the entry have an appropriate value. Junos OS supports only 1+1 architecture.</p> <p>The values 1 through 14 define a working channel. When an attempt is made to set the corresponding <code>apsConfigRowStatus</code> field to active, the <code>apsChanConfigNumber</code> values of all entries with equal <code>apsChanConfigGroupName</code> fields must be a set of consecutive integer values beginning with 0 or 1, depending on the architecture of the group, and ending with <i>n</i>, where <i>n</i> is greater than or equal to 1 and less than or equal to 14. Otherwise, the error <code>inconsistentValue</code> is returned to the <code>apsConfigRowStatus</code> set attempt.</p>

Table 345: apsChanConfigEntry (continued)

Object	Object Identifier	Description
apsChanConfigIfIndex	apsChanConfigEntry 4	<p>The interface index assigned to a SONET LTE. This is an interface with ifType sonet(39). The value of this object must be unique among all instances of apsChanConfigIfIndex. In other words, a particular SONET LTE can only be configured in one APS group.</p> <p>This object cannot be set if the apsChanConfigGroupName instance associated with this row is equal to an instance of apsConfigName and the corresponding apsConfigRowStatus object is set to active. In other words, this value cannot be changed if the APS group is active. However, this value can be changed if the apsConfigRowStatus value is equal to notInService. Junos OS supports only read access.</p>
apsChanConfigPriority	apsChanConfigEntry 5	The priority of the channel. This field returns the value low priority. Junos OS supports only 1+1 architecture.
apsChanConfigStorageType	apsChanConfigEntry 6	The storage type for this conceptual row. Conceptual rows having the value permanent need not allow write access to any columnar objects in the row. For information about conceptual rows, see RFC 2579, <i>Textual Conventions for SMIv2</i> .

apsChanStatusTable

apsChanStatusTable provides APS channel statistics.

apsChanStatusEntry objects have read access only and are listed in [Table 346 on page 810](#).

Table 346: apsChanStatusTable

Object	Object Identifier	Description
apsChanStatusCurrent	apsChanStatusEntry 1	<p>The current state of the port. This object has the following values:</p> <p>lockedOut—This bit, when applied to a working channel, indicates that the channel is prevented from switching to the protection line. When applied to the null channel, this bit indicates that no working channel can switch to the protection line.</p> <p>sd—A signal degrade condition is in effect.</p> <p>sf—A signal failure condition is in effect switched. The switched bit is applied to a working channel if that channel is currently switched to the protection line.</p> <p>wtr—A wait-to-restore state is in effect.</p>
apsChanStatusSignalDegrades	apsChanStatusEntry 2	<p>A count of signal degrade conditions. A signal degrade condition occurs when the line bit error rate (BER) exceeds the currently configured value of the relevant instance of apsConfigSdBerThreshold. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.</p>
apsChanStatusSignalFailures	apsChanStatusEntry 3	<p>A count of signal failure conditions that have been detected on the incoming signal. A signal failure condition occurs when a loss of signal, loss of frame, AIS-L or line bit error rate exceeds the currently configured value of the relevant instance of apsConfigSfBerThreshold. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.</p>

Table 346: apsChanStatusTable (continued)

Object	Object Identifier	Description
apsChanStatusSwitchovers	apsChanStatusEntry 4	<p>When queried with index value apsChanConfigNumber other than 0, this object returns the number of times this channel has switched to the protection line.</p> <p>When queried with index value s set to 0, which is the protection line, this object returns the number of times that any working channel has switched back to the working line from this protection line. Discontinuities in the value of this counter can occur when the management system is reinitialized, and at other times as indicated by the value of apsChanStatusDiscontinuityTime.</p>
apsChanStatusLastSwitchover	apsChanStatusEntry 5	<p>When queried with index value apsChanConfigNumber other than 0, this object returns the value of sysUpTime when this channel last completed a switch to the protection line. If this channel has never switched to the protection line, the value 0 is returned.</p> <p>When queried with index value apsChanConfigNumber set to 0, which is the protection line, this object returns the value of sysUpTime the last time that a working channel was switched back to the working line from this protection line. If no working channel has ever switched back to the working line from this protection line, the value 0 is returned.</p>

Table 346: `apsChanStatusTable` (continued)

Object	Object Identifier	Description
<code>apsChanStatusSwitchoverSeconds</code>	<code>apsChanStatusEntry</code> 6	<p>The cumulative Protection Switching Duration (PSD) time, in seconds. For a working channel, this is the cumulative number of seconds that service was carried on the protection line. For the protection line, this is the cumulative number of seconds that the protection line has been used to carry any working channel traffic.</p> <p>This information is only valid if revertive switching is enabled. The value 0 is returned. Otherwise, discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of <code>apsChanStatusDiscontinuityTime</code>. For example, if the value of an instance of <code>apsChanStatusSwitchoverSeconds</code> changes from a nonzero value to zero due to revertive switching being disabled. It is expected that the corresponding value of <code>apsChanStatusDiscontinuityTime</code> is updated to reflect the time of the configuration change.</p>
<code>apsChanStatusDiscontinuityTime</code>	<code>apsChanStatusEntry</code> 7	<p>The value of <code>sysUpTime</code> on the most recent occasion at which any one or more of this channel's counters suffered a discontinuity. The relevant counters are the specific instances associated with this channel of any <code>Counter32</code> object contained in <code>apsChanStatusTable</code>. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a zero value for <code>apsChanStatusEntry</code>.</p>

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Source Class Usage MIB

- [Interpreting the Enterprise-Specific Source Class Usage MIB on page 813](#)

Interpreting the Enterprise-Specific Source Class Usage MIB

- [Source Class Usage MIB on page 813](#)
- [jnxScuStatsTable on page 813](#)

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 *Routing Policy Feature Guide for Routing Devices* and the *Junos OS Network Interfaces Library for Routing Devices*. For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/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 813](#)

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 347 on page 814](#).

Table 347: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

SPU Monitoring Objects MIB

- [Interpreting the Enterprise-Specific SPU Monitoring Objects MIB on page 815](#)

Interpreting the Enterprise-Specific SPU Monitoring Objects MIB

- [SPU Monitoring Objects MIB on page 815](#)
- [SPU Monitoring Objects Table on page 816](#)
- [Cluster Objects Table on page 817](#)

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/junos13.2/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 816](#)
- [Cluster Objects Table on page 817](#)

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 348 on page 816](#) describes the SPU monitoring objects.

Table 348: SPU Monitoring Objects Table

Object	Object ID	Description
<code>JnxJsSPUMonitoringObjectsEntry</code>	<code>jnxJsSPUMonitoringObjectsTable 1</code>	<p>Each entry collects information about the memory utilization for an SPU.</p> <p>Indexes:</p> <ul style="list-style-type: none"> • <code>jnxJsSPUMonitoringIndex</code> • <code>jnxJsSPUMonitoringFPCIndex</code> • <code>jnxJsSPUMonitoringSPUIndex</code> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • <code>jnxJsSPUMonitoringCPUUsage</code> • <code>jnxJsSPUMonitoringMemoryUsage</code> • <code>jnxJsSPUMonitoringCurrentFlowSession</code> • <code>jnxJsSPUMonitoringMaxFlowSession</code> • <code>jnxJsSPUMonitoringCurrentCPSession</code> • <code>jnxJsSPUMonitoringMaxCPSession</code> • <code>jnxJsSPUMonitoringMaxTotalSession</code> • <code>jnxJsSPUMonitoringNodeIndex</code> • <code>jnxJsSPUMonitoringNodeDescr</code>
<code>jnxJsSPUMonitoringIndex</code>	<code>jnxJsSPUMonitoringObjectsEntry 1</code>	Indicates an SPU's overall index in the system.
<code>jnxJsSPUMonitoringFPCIndex</code>	<code>jnxJsSPUMonitoringObjectsEntry 2</code>	Indicates which FPC the SPU is on .
<code>jnxJsSPUMonitoringSPUIndex</code>	<code>jnxJsSPUMonitoringObjectsEntry 3</code>	Indicates the index of an SPU inside the FPC.
<code>jnxJsSPUMonitoringCPUUsage</code>	<code>jnxJsSPUMonitoringObjectsEntry 4</code>	Indicates the current utilization percentage of an SPU.
<code>jnxJsSPUMonitoringMemoryUsage</code>	<code>jnxJsSPUMonitoringObjectsEntry 5</code>	Indicates the current percentage of memory usage of an SPU.
<code>jnxJsSPUMonitoringCurrentFlowSession</code>	<code>jnxJsSPUMonitoringObjectsEntry 6</code>	Indicates the current flow sessions of an SPU.
<code>jnxJsSPUMonitoringMaxFlowSession</code>	<code>jnxJsSPUMonitoringObjectsEntry 7</code>	Indicates the maximum flow sessions of an SPU.
<code>jnxJsSPUMonitoringCurrentCPSession</code>	<code>jnxJsSPUMonitoringObjectsEntry 8</code>	Indicates the current number of central point (CP) sessions on an SPU.

Table 348: SPU Monitoring Objects Table (*continued*)

Object	Object ID	Description
<code>jnxJsSPUMonitoringMaxCPSession</code>	<code>jnxJsSPUMonitoringObjectsEntry 9</code>	Indicates the maximum number of CP sessions on an SPU.
<code>jnxJsSPUMonitoringMaxTotalSession</code>	<code>jnxJsSPUMonitoringObjectsEntry 10</code>	Indicates the maximum number of sessions on the device.
<code>jnxJsSPUMonitoringNodeIndex</code>	<code>jnxJsSPUMonitoringObjectsEntry 11</code>	Identifies a chassis in single or cluster mode.
<code>jnxJsSPUMonitoringNodeDescr</code>	<code>jnxJsSPUMonitoringObjectsEntry 12</code>	Describes the chassis cluster node information.

Cluster Objects Table

The object identifier for the `jnxJsClusterObjectsTable` is `{jnxJsSPUMonitoringMIB 4}`. The cluster objects provide statistical information for each node. [Table 349 on page 817](#) describes the cluster objects.

Table 349: Cluster Objects Table

Object	Object ID	Description
<code>jnxJsClusterObjectsEntry</code>	<code>jnxJsSPUMonitoringObjectsTable 1</code>	Each entry collects information per node.
<code>JnxJsClusterObjectsEntry</code>	—	Index: <ul style="list-style-type: none"> <code>jnxJsClusterMonitoringNodeIndex</code> Sequence of parameters: <ul style="list-style-type: none"> <code>jnxJsClusterMonitoringNodeDescr</code> <code>jnxJsNodeCurrentTotalSession</code> <code>jnxJsNodeMaxTotalSession</code> <code>jnxJsNodeSessionCreationPerSecond</code>
<code>jnxJsClusterMonitoringIndex</code>	<code>jnxJsClusterObjectsEntry 1</code>	Indicates the index number of the node.
<code>jnxJsClusterMonitoringNodeDescr</code>	<code>jnxJsClusterObjectsEntry 2</code>	Describes the node being monitored.
<code>jnxJsNodeCurrentTotalSession</code>	<code>jnxJsClusterObjectsEntry 3</code>	Indicates the total sessions currently running on the node.
<code>jnxJsNodeMaxTotalSession</code>	<code>jnxJsClusterObjectsEntry 4</code>	Indicates the maximum number of sessions that can run on the node.
<code>jnxJsNodeSessionCreationPerSecond</code>	<code>jnxJsClusterObjectsEntry 5</code>	Indicates the number of sessions created per second on the node.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

Structure of Management Information MIB

- Interpreting the Enterprise-Specific Structure of Management Information MIB on page 819

Interpreting the Enterprise-Specific Structure of Management Information MIB

- Structure of Management Information MIB on page 819
- jnxProducts on page 819
- jnxServices on page 820
- jnxMibs on page 821
- jnxTraps on page 822
- jnxExperiment on page 823

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/junos13.2/topics/reference/mibs/mib-jnx-smi.txt.

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

- jnxProducts on page 819
- jnxServices on page 820
- jnxMibs on page 821
- jnxTraps on page 822
- jnxExperiment on page 425

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.

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}**.

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 25](#).

jnxMibs

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

- MPLS MIB—Whose object identifier is **{jnxMibs 2}**.
- Juniper Networks enterprise-specific extensions to the Interface MIB—Whose object identifier is **{jnxMibs 3}**.
- Alarm MIB—Whose object identifier is **{jnxMibs 4}**.
- Firewall MIB—Whose object identifier is **{jnxMibs 5}**.
- Destination Class Usage MIB—Whose object identifier is **{jnxMibs 6}**.
- Juniper Networks enterprise-specific extensions to the Ping MIB—Whose object identifier is **{jnxMibs 7}**.
- Juniper Networks enterprise-specific extensions to the Traceroute MIB—Whose object identifier is **{jnxMibs 8}**.
- ATM MIB—Whose object identifier is **{jnxMibs 10}**.
- IPv6 and ICMPv6 MIB—Whose object identifier is **{jnxMibs 11}**.
- IPv4 MIB—Whose object identifier is **{jnxMibs 12}**.
- Juniper Networks enterprise-specific extensions to the RMON Events and Alarms MIB—Whose object identifier is **{jnxMIBs 13}**.
- Juniper Networks enterprise-specific extensions to the LDP Traps MIB—Whose object identifier is **{jnxMibs 14}**.
- Class-of-Service MIB—Whose object identifier is **{jnxMibs 15}**.
- Source Class Usage MIB—Whose object identifier is **{jnxMibs 16}**.
- Reverse-Path-Forwarding MIB—Whose object identifier is **{jnxMibs 17}**.
- Configuration Management MIB—Whose object identifier is **{jnxMibs 18}**.
- Passive Monitoring MIB—Whose object identifier is **{jnxMibs 19}**.
- SONET/SDH Interface Management MIB—Whose object identifier is **{jnxMibs 20}**.
- ATM Class-of-Service MIB—Whose object identifier is **{jnxMibs 21}**.
- IPsec Monitoring MIB—Whose object identifier is **{jnxMibs 22}**.
- Ethernet MAC MIB—Whose object identifier is **{jnxMibs 23}**.

- SONET APS MIB—Whose object identifier is **{jnxMibs 24}**.
- Chassis Definitions for Router Model MIB—Whose object identifier is **{jnxMibs 25}**.
- VPN MIB—Whose object identifier is **{jnxMibs 26}**.
- Flow Collection Services MIB—Whose object identifier is **{jnxMibs 28}**.
- RSVP Traffic Engineering (TE) MIB—Whose object identifier is **{jnxMibs 30}**.
- Host Resources MIB—Whose object identifier is **{jnxMibs 31}**.
- Services PIC MIB—Whose object identifier is **{jnxMibs 32}**.
- Dynamic Flow Capture (DFC) MIB—Whose object identifier is **{jnxMibs 33}**.
- Chassis Forwarding MIB—Whose object identifier is **{jnxMibs 34}**.
- System Log MIB—Whose object identifier is **{jnxMibs 35}**.
- MPLS LDP MIB—Whose object identifier is **{jnxMibs 36}**.
- Event MIB—Whose object identifier is **{jnxMibs 37}**.
- IP Forward MIB—Whose object identifier is **{jnxMibs 38}**.
- **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}**.
- 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 25](#).

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}**.
- **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}**.

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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)

- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 76

Subscriber MIB

- Interpreting the Enterprise-Specific Subscriber MIB on page 825

Interpreting the Enterprise-Specific Subscriber MIB

- Subscriber MIB on page 825
- Subscriber MIB Textual Conventions on page 826
- Managed Objects for General Subscriber Functions and `jnxSubscriberTable` on page 826
- Managed Objects for Logical System Subscriber Information and `jnxSubscriberLogicalSystemTable` on page 828
- Managed Objects for Routing Instance Subscriber Information and `jnxSubscriberRoutingInstanceTable` on page 829

Subscriber MIB

The Juniper Networks enterprise-specific Subscriber MIB, whose object ID is **{jnxSubscriberMibRoot 1}**, defines an object to enable retrieval of subscriber-related information. Subscriber state information is maintained in the session database. Using the object, you can view subscriber information using SNMP **get** and **get-next** requests.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-subscriber.txt.

For information about the enterprise-specific Subscriber MIB objects, see the following topics:

- Subscriber MIB Textual Conventions on page 826
- Managed Objects for General Subscriber Functions and `jnxSubscriberTable` on page 826
- Managed Objects for Logical System Subscriber Information and `jnxSubscriberLogicalSystemTable` on page 828
- Managed Objects for Routing Instance Subscriber Information and `jnxSubscriberRoutingInstanceTable` on page 829

Subscriber MIB Textual Conventions

Table 350 on page 826 contains information about the textual conventions used in the enterprise-specific Subscriber MIB.

Table 350: Subscriber MIB Textual Conventions

Object	Description	Syntax
jnxSubscriberState	State of the AAA subscriber.	This object uses the following integer values: <ul style="list-style-type: none"> • 0—INIT state only • 1—Configured state only • 2—Active state only • 3—Terminating state only • 4—Terminated state only • 5—Unknown state
jnxSubscriberClientType	Type of subscriber client.	This object uses the following integer values: <ul style="list-style-type: none"> • 0—None • 1—DHCP • 2—VLAN • 3—Generic • 4—MobileIp • 5—vplsPw • 6—PPP • 7—PPPoE • 8—L2TP • 9—Static • 10—MLPPP

Managed Objects for General Subscriber Functions and jnxSubscriberTable

The managed objects listed in Table 351 on page 826 along with the objects in **jnxSubscriberTable** provide general subscriber information such as the number of active subscribers and the total number of subscribers.

Each **jnxSubscriberGeneral**, whose object identifier is **{jnxSubscriberObjects 1}**, contains the objects listed in Table 351 on page 826.

Table 351: Managed Objects for General Subscriber Functions

Object	Object ID	Description
jnxSubscriberTotalCount	jnxSubscriberGeneral 1	Total number of subscribers.
jnxSubscriberActiveCount	jnxSubscriberGeneral 2	Number of active subscribers.

For information about **jnxSubscriberTable**, see the following section:

- [jnxSubscriberTable](#) on page 827

jnxSubscriberTable

jnxSubscriberTable, whose object identifier is **{jnxSubscriberGeneral 3}**, contains **jnxSubscriberEntry** that maps to the specification of each subscriber.

Each **jnxSubscriberEntry**, whose object identifier is **{jnxSubscriberTable 1}**, contains the objects listed in [Table 352 on page 827](#).

Table 352: jnxSubscriberTable

Object	Object ID	Description
jnxSubscriberHandleHiWord	jnxSubscriberEntry 1	Subscriber handle associated with each subscriber. Returns the most significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
jnxSubscriberHandleLoWord	jnxSubscriberEntry 2	Subscriber handle associated with each subscriber. Returns the least significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
jnxSubscriberUserName	jnxSubscriberEntry 3	Name of the user associated with the subscriber.
jnxSubscriberClientType	jnxSubscriberEntry 4	Type of the client associated with the subscriber.
jnxSubscriberIpAddress	jnxSubscriberEntry 5	Client IP address assigned to the subscriber.
jnxSubscriberIpAddressMask	jnxSubscriberEntry 6	Client IP address mask assigned to the subscriber.
jnxSubscriberLogicalSystem	jnxSubscriberEntry 7	Logical system associated with the subscriber.
jnxSubscriberRoutingInstance	jnxSubscriberEntry 8	Routing instance associated with the subscriber.
jnxSubscriberInterface	jnxSubscriberEntry 9	Textual name of the interface associated with the subscriber.
jnxSubscriberInterfaceType	jnxSubscriberEntry 10	Type of interface associated with the subscriber. Possible values include: <ul style="list-style-type: none"> • none (0) • static (1) • dynamic (2)
jnxSubscriberMacAddress	jnxSubscriberEntry 11	MAC address of the interface associated with the subscriber.
jnxSubscriberState	jnxSubscriberEntry 12	State of the subscriber.
jnxSubscriberLoginTime	jnxSubscriberEntry 13	System time when the subscriber logged in to the server.

Table 352: jnxSubscriberTable (*continued*)

Object	Object ID	Description
jnxSubscriberAcctSessionId	jnxSubscriberEntry 14	Accounting session ID associated with the subscriber.

Managed Objects for Logical System Subscriber Information and jnxSubscriberLogicalSystemTable

The managed objects listed in [Table 353 on page 828](#) along with the objects in **jnxSubscriberLogicalSystemTable** provide subscriber information based on the associated logical system, such as the number of active subscribers on the logical system and the total number of subscribers on the logical system.

Each **jnxSubscriberLogicalSystemObjects**, whose object identifier is **{jnxSubscriberObjects 2}**, contains the objects listed in [Table 353 on page 828](#).

Table 353: Managed Objects for Logical System Subscriber Functions

Object	Object ID	Description
jnxSubscriberLogicalSystemTotalCount	jnxSubscriberLogicalSystemObjects 1	Total number of subscribers on the logical system.
jnxSubscriberLogicalSystemActiveCount	jnxSubscriberLogicalSystemObjects 2	Number of active subscribers on the logical system.

For information about **jnxSubscriberLogicalSystemTable**, see the following section:

- [jnxSubscriberLogicalSystemTable on page 828](#)

jnxSubscriberLogicalSystemTable

jnxSubscriberLogicalSystemTable, whose object identifier is **{jnxSubscriberLogicalSystemObjects 3}**, contains **jnxSubscriberLogicalSystemEntry** that maps to the specification of each subscriber. You must provide the name of the logical system in the SNMP query. The logical system is identified either by the context field in V3 requests or by the community string in V1 and V2c requests. You can query for information specific to the logical system or routing instance defined within a logical system by using the prefix **<logical-system>/<routing-instance> @** with the community string. In the prefix, **@** acts as a separator between the community string and the name of the routing instance, and **/** acts as a separator between the logical system name and the routing instance. If you do not provide the name of the logical system, the default logical system data is displayed. The name of the routing instance is ignored for this table.

Each **jnxSubscriberLogicalSystemEntry**, whose object identifier is **{jnxSubscriberLogicalSystemTable 1}**, contains the objects listed in [Table 354 on page 829](#).

Table 354: jnxSubscriberLogicalSystemTable

Object	Object ID	Description
jnxSubscriberLogicalSystemHandleHiWord	jnxSubscriberLogicalSystemEntry 1	Subscriber handle associated with each subscriber. Returns the most significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
jnxSubscriberLogicalSystemHandleLoWord	jnxSubscriberLogicalSystemEntry 2	Subscriber handle associated with each subscriber. Returns the least significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
jnxSubscriberLogicalSystemState	jnxSubscriberLogicalSystemEntry 3	State of the subscriber.

Managed Objects for Routing Instance Subscriber Information and jnxSubscriberRoutingInstanceTable

The managed objects listed in [Table 355 on page 829](#) along with the objects in **jnxSubscriberRoutingInstanceTable** provide subscriber information based on the associated routing instance, such as the number of active subscribers on the routing instance and the total number of subscribers on the routing instance.

Each **jnxSubscriberRoutingInstanceObjects**, whose object identifier is **{jnxSubscriberObjects 3}**, contains the objects listed in [Table 355 on page 829](#).

Table 355: Managed Objects for Routing Instance Subscriber Functions

Object	Object ID	Description
jnxSubscriberRoutingInstanceTotalCount	jnxSubscriberRoutingInstanceObjects 1	Total number of subscribers on the routing instance.
jnxSubscriberRoutingInstanceActiveCount	jnxSubscriberRoutingInstanceObjects 2	Number of active subscribers on the routing instance.

For information about **jnxSubscriberRoutingInstanceTable**, see the following section:

- [jnxSubscriberRoutingInstanceTable on page 829](#)

jnxSubscriberRoutingInstanceTable

jnxSubscriberRoutingInstanceTable, whose object identifier is **{jnxSubscriberRoutingInstanceObjects 3}**, contains **jnxSubscriberRoutingInstanceEntry** that maps to the specification of each subscriber. You must provide the name of the routing instance in the SNMP query. The routing instance is identified either by the context field in V3 requests or by the community string in V1 and V2c requests. You can query for information specific to the routing instance or routing instance defined within a logical system by using the prefix **<logical-system>/<routing-instance> @** with the community

string. In the prefix, @ acts as a separator between the community string and the name of the routing instance, and / acts as a separator between the logical system name and the routing instance. If you do not provide the name of the routing instance, the default routing instance data is displayed. The name of the logical system is ignored for this table.

Each `jnxSubscriberRoutingInstanceEntry`, whose object identifier is `{jnxSubscriberRoutingInstanceTable 1}`, contains the objects listed in [Table 356 on page 830](#).

Table 356: jnxSubscriberRoutingInstanceTable

Object	Object ID	Description
<code>jnxSubscriberRoutingInstanceHandleHiWord</code>	<code>jnxSubscriberRoutingInstanceEntry 1</code>	Subscriber handle associated with each subscriber. Returns the most significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
<code>jnxSubscriberRoutingInstanceHandleLoWord</code>	<code>jnxSubscriberRoutingInstanceEntry 2</code>	Subscriber handle associated with each subscriber. Returns the least significant 32 bits of the 64-bit subscriber ID. The value of the subscriber handle is a monotonically increasing number.
<code>jnxSubscriberRoutingInstanceState</code>	<code>jnxSubscriberRoutingInstanceEntry 3</code>	State of the subscriber.

- Related Documentation**
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
 - [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

System Log MIB

- Interpreting the Enterprise-Specific System Log MIB on page 831

Interpreting the Enterprise-Specific System Log MIB

- System Log MIB on page 831
- jnxSyslogTable on page 831
- jnxSyslogAvTable on page 834

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/junos13.2/topics/reference/mibs/mib-jnx-syslog.txt.

For information about the enterprise-specific System Log MIB objects, see the following topics:

- jnxSyslogTable on page 831
- jnxSyslogAvTable on page 834

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 357 on page 832](#). Each entry provides information about a single system log message.

Table 357: 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: <ul style="list-style-type: none"> • 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 358 on page 833 .)
jnxSyslogProcessID	jnxSyslogEntry 6	Process ID of the process that generated the system log message.
jnxSyslogProcessName	jnxSyslogEntry 7	Process that generated the system log message.
jnxSyslogHostName	jnxSyslogEntry 8	Hostname of the machine that generated the system log.
jnxSyslogMessage	jnxSyslogEntry 9	System log message that was generated.

[Table 358 on page 833](#) lists the facilities that generate system log messages.

Table 358: 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
24	interact	Junos names

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 359 on page 834](#). Each entry provides information about attribute value pairs of system log messages generated by a device.

Table 359: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 78

Traceroute MIB

- [Interpreting the Enterprise-Specific Traceroute MIB on page 835](#)

Interpreting the Enterprise-Specific Traceroute MIB

- [Traceroute MIB on page 835](#)
- [jnxTraceRouteCtlTable on page 835](#)

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/junos13.2/topics/reference/mibs/mib-jnx-traceroute.txt.

For more information about using the Traceroute MIB and enterprise-specific Traceroute MIB, see the *Network Management Administration Guide for Routing Devices*.

For information about the enterprise-specific Traceroute MIB objects, see the following topic:

- [jnxTraceRouteCtlTable on page 835](#)

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 360 on page 836](#).

Table 360: 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	<p>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.</p> <p>The precedence order is as follows:</p> <ul style="list-style-type: none"> • traceRouteCtlIfIndex (see traceRouteCtlTable in the “Traceroute MIB” on page 835) • 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 79

Utility MIB

- [Interpreting the Enterprise-Specific Utility MIB on page 837](#)

Interpreting the Enterprise-Specific Utility MIB

- [Utility MIB on page 837](#)
- [jnxUtilCounter32Table on page 838](#)
- [jnxUtilCounter64Table on page 838](#)
- [jnxUtilIntegerTable on page 838](#)
- [jnxUtilUintTable on page 839](#)
- [jnxUtilStringTable on page 839](#)

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/junos13.2/topics/reference/mibs/mib-jnx-util.txt.

For information about the enterprise-specific Utility MIB objects, see the following topics:

- [jnxUtilCounter32Table on page 838](#)
- [jnxUtilCounter64Table on page 838](#)
- [jnxUtilIntegerTable on page 838](#)

- [jnxUtilUintTable](#) on page 839
- [jnxUtilStringTable](#) on page 839

jnxUtilCounter32Table

jnxUtilCounter32Table, whose object ID is **{jnxUtilData 1}**, contains information about 32-bit counters.

Each **jnxUtilCounter32Entry** has the objects listed in [Table 361](#) on page 838.

Table 361: jnxUtilCounter32Entry

Object	Object ID	Description
jnxUtilCounter32Name	jnxUtilCounter32Entry 1	Name assigned to the instance.
jnxUtilCounter32Value	jnxUtilCounter32Entry 2	Value of the instance.
jnxUtilCounter32Time	jnxUtilCounter32Entry 3	Time when the instance was last populated.

jnxUtilCounter64Table

jnxUtilCounter64Table, whose object ID is **{jnxUtilData 2}**, contains information about 64-bit counters.

Each **jnxUtilCounter64Entry** has the objects listed in [Table 362](#) on page 838.

Table 362: jnxUtilCounter64Entry

Object	Object ID	Description
jnxUtilCounter64Name	jnxUtilCounter64Entry 1	Name assigned to the instance.
jnxUtilCounter64Value	jnxUtilCounter64Entry 2	Value of the instance.
jnxUtilCounter64Time	jnxUtilCounter64Entry 3	Time when the instance was last populated.

jnxUtilIntegerTable

jnxUtilIntegerTable, whose object ID is **{jnxUtilData 3}**, contains information about signed integer values.

Each **jnxUtilIntegerEntry** contains the objects listed in [Table 363](#) on page 838.

Table 363: jnxUtilIntegerEntry

Object	Object ID	Description
jnxUtilIntegerName	jnxUtilIntegerEntry 1	Name assigned to the instance.
jnxUtilIntegerValue	jnxUtilIntegerEntry 2	Value of the instance.

Table 363: jnxUtilIntegerEntry (*continued*)

Object	Object ID	Description
jnxUtilIntegerTime	jnxUtilIntegerEntry 3	Time when the instance was last populated.

jnxUtilUintTable

jnxUtilUintTable, whose object ID is **{jnxUtilData 4}**, contains information about unsigned integer values.

Each **jnxUtilUintEntry** has the objects listed in [Table 364 on page 839](#).

Table 364: 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.

jnxUtilStringTable

jnxUtilStringTable, whose object ID is **{jnxUtilData 5}**, contains information about octet strings.

Each **jnxUtilStringEntry** contains the objects listed in [Table 365 on page 839](#).

Table 365: 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

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 80

VLAN MIB

- Interpreting the Enterprise-Specific VLAN MIB on page 841

Interpreting the Enterprise-Specific VLAN MIB

- VLAN MIB on page 841
- VLAN Configuration Table on page 842
- VLAN Interface Table on page 843
- Port Group Table on page 844
- MAC List Table on page 846

VLAN MIB

The Juniper Networks enterprise-specific VLAN MIB for EX Series Ethernet Switches, whose object identifier is **{jnxExSwitching 5}**, contains information about prestandard IEEE 802.10 VLANs and their association with LAN Emulation Clients (LAC). Devices with prestandard implementation maintain port groupings and associated filters that are used to form a virtual bridge.

The enterprise-specific VLAN MIB leverages the following objects and data types from standard MIBs, RFCs, and Juniper Networks enterprise-specific MIBs:

- **Integer 32** and **IpAddress**—From SNMPv2—SMI
- **MacAddress**, **DisplayString**, and **TruthValue**—From SNMPv2—TC
- **InterfaceIndex**—From IF MIB
- **InetAddress** and **InetAddressType**—From Inet Address MIB
- **jnxExVlan**—From Juniper Networks enterprise-specific SMI MIB

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vlan.txt.



NOTE: **jnxVlanTable**, **jnxVlanInterfaceTable**, and **jnxVlanPortGroupTable** have been deprecated and are replaced by **jnxExVlanTable**, **jnxExVlanInterfaceTable**, and **jnxExVlanPortGroupTable**.

For information about the enterprise-specific VLAN MIB objects, see the following topics:

- [VLAN Configuration Table on page 842](#)
- [VLAN Interface Table on page 843](#)
- [Port Group Table on page 844](#)
- [MAC List Table on page 846](#)

VLAN Configuration Table



NOTE: `jnxVlanTable` has been deprecated and is replaced by `jnxExVlanTable`. See [“jnxExVlanTable” on page 842](#).

The `jnxVlanTable`, whose object identifier is `{jnxVlanMibObjects 1}`, contains VLAN names and properties. Each `jnxVlanEntry` contains the objects listed in [Table 366 on page 842](#).

Table 366: jnxVlanTable

Object	Object ID	Description
<code>jnxVlanName</code>	<code>jnxVlanEntry 1</code>	Contains the name of the VLAN, which can be a string of not more than 255 characters.
<code>jnxVlanID</code>	<code>jnxVlanEntry 2</code>	Contains the identifier used internally by the device to reference the VLAN. This object can be an integer value in the range 1 through 4094.
<code>jnxVlanType</code>	<code>jnxVlanEntry 3</code>	Denotes the type of the VLAN. This object uses the following integer values: <ul style="list-style-type: none"> • 1—Static • 2—Dynamic The default value for this object is 1, static.
<code>jnxVlanPortGroupInstance</code>	<code>jnxVlanEntry 4</code>	Contains the index that identifies the subtree in the <code>jnxVlanPortGroupTable</code> .
<code>jnxVlanMacListInstance</code>	<code>jnxVlanEntry 5</code>	Contains an index that identifies the subtree to retrieve the list of MAC addresses to the <code>jnxVlanMacListTable</code> subtree to retrieve in.

`jnxExVlanTable`

`jnxExVlanTable`, whose object identifier is `{jnxVlanMibObjects 5}`, replaces the deprecated `jnxVlanTable` and contains the objects listed in [Table 367 on page 843](#).

Table 367: jnxExVlanTable

Object	Object ID	Description
jnxExVlanID	jnxExVlanEntry 1	Contains the identifier used internally by the device to reference the VLAN. This object can be an integer value in the range 1 through 4094.
jnxExVlanName	jnxExVlanEntry 2	Contains the name of the VLAN. VLAN name can be a string of not more than 255 characters.
jnxExVlanType	jnxExVlanEntry 3	Denotes the type of the VLAN. This object uses the following integer values: <ul style="list-style-type: none"> 1—Static 2—Dynamic The default value for this object is 1, static.
jnxExVlanPortGroupInstance	jnxExVlanEntry 4	Contains the index that identifies the subtree in the jnxExVlanPortGroupTable.

VLAN Interface Table



NOTE: jnxVlanInterfaceTable has been deprecated and is replaced by jnxExVlanInterfaceTable. See “jnxExVlanInterfaceTable” on page 844.

The jnxVlanInterfaceTable, whose object identifier is {jnxVlanMIBObjects 2}, contains information about the Layer 3 properties of VLANs. Each jnxVlanInterfaceEntry, indexed with jnxVlanName, contains the objects listed in Table 368 on page 843.

Table 368: jnxVlanInterfaceTable

Object	Object ID	Description
jnxVlanInterfaceIpAddress	jnxVlanInterfaceEntry 1	Contains the IP address of the interface.
jnxVlanInterfaceProtocol	jnxVlanInterfaceEntry 2	Specifies the protocol used.
jnxVlanInterfaceSubNetMask	jnxVlanInterfaceEntry 3	Specifies the subnet mask of the VLAN,
jnxVlanInterfaceBroadcastAddress	jnxVlanInterfaceEntry 4	Specifies the broadband address of the VLAN.
jnxVlanInterfaceDescription	jnxVlanInterfaceEntry 5	Contains a description for the VLAN. This object can contain a string of not more than 255 characters.
jnxVlanInterfaceAdminStatus	jnxVlanInterfaceEntry 6	Denotes the administration status of the VLAN.
jnxVlanInterfaceOperStatus	jnxVlanInterfaceEntry 7	Denotes the operational status of the VLAN.
jnxVlanSnmplfIndex	jnxVlanInterfaceEntry 8	Specifies the SNMP IF Index for the interface.

jnxExVlanInterfaceTable

jnxExVlanInterfaceTable, whose object identifier is {**jnxVlanMIBObjects 6**}, replaces the deprecated **jnxVlanInterfaceTable** and contains the objects listed in [Table 369 on page 844](#).

Table 369: jnxExVlanInterfaceTable

Object	Object ID	Description
jnxExVlanInterfaceProtocol	jnxExVlanInterfaceEntry 1	Specifies the protocol used.
jnxExVlanInterfaceIpAddress	jnxExVlanInterfaceEntry 2	Contains the IP address of the interface.
jnxExVlanInterfacePrefixLength	jnxExVlanInterfaceEntry 3	Specifies the subnet mask of the VLAN,
jnxExVlanInterfaceBroadcastAddress	jnxExVlanInterfaceEntry 4	Specifies the broadband address of the VLAN.
jnxExVlanInterfaceDescription	jnxExVlanInterfaceEntry 5	Contains a description for the VLAN. This object can contain a string of not more than 255 characters.
jnxExVlanInterfaceAdminStatus	jnxExVlanInterfaceEntry 6	Denotes the administration status of the VLAN.
jnxExVlanInterfaceOperStatus	jnxExVlanInterfaceEntry 7	Denotes the operational status of the VLAN.
jnxExVlanSnmplfIndex	jnxExVlanInterfaceEntry 8	Specifies the SNMP IF Index for the interface.

Port Group Table



NOTE: **jnxVlanPortGroupTable** has been deprecated and is replaced by **jnxExVlanPortGroupTable**. See “[jnxExVlanPortGroupTable](#)” on page 845.

The **jnxVlanPortGroupTable** contains information about port groupings. Each **jnxVlanPortGroupEntry** contains the objects listed in [Table 370 on page 844](#).

Table 370: jnxVlanPortGroupTable

Object	Object ID	Description
jnxVlanPortGroupIndex	jnxVlanPortGroupEntry 1	Uniquely identifies a port group.
jnxVlanPort	jnxVlanPortGroupEntry 2	Specifies the port on the VLAN with which this port group is associated.

Table 370: jnxVlanPortGroupTable (*continued*)

Object	Object ID	Description
jnxVlanPortStatus	jnxVlanPortGroupEntry 3	<p>Shows the status of association between the port and the VLAN. This object uses the following integer values:</p> <ul style="list-style-type: none"> • 1—autoActive: The port is part of the VLAN because the switch has automatically added the port. • 2—allowed: The port has been configured to be part of the VLAN, and is allowed to be part of the VLAN, if the port meets all other requirements. • 3—allowedActive: The port has been configured to be part of the VLAN, and is allowed to be part of the VLAN, if the port meets all other requirements. However, unlike in the case of allowed ports, this port has a device that is participating in the VLAN associated with the port. • 4—allowedNotAvail: The port is active on some other VLAN, and is not available currently. This value applies to devices that do not allow a port to be part of more than one VLAN at the same time. • 5—notAssociated: The port is part of a port group that is not associated with the VLAN.

jnxExVlanPortGroupTable

jnxExVlanPortGroupTable replaces the deprecated **jnxVlanPortGroupTable** and contains the objects listed in [Table 371 on page 845](#).

Table 371: jnxExVlanPortGroupTable

Object	Object ID	Description
jnxExVlanPortGroupIndex	jnxExVlanPortGroupEntry 1	Uniquely identifies a port group.
jnxExVlanPort	jnxExVlanPortGroupEntry 2	Specifies the port on the VLAN with which this port group is associated.

Table 371: jnxExVlanPortGroupTable (*continued*)

Object	Object ID	Description
jnxExVlanPortStatus	jnxExVlanPortGroupEntry 3	<p>Shows the status of association between the port and the VLAN. This object uses the following integer values:</p> <ul style="list-style-type: none"> • 1—autoActive: The port is part of the VLAN because the switch has automatically added the port. • 2—allowed: The port has been configured to be part of the VLAN, and is allowed to be part of the VLAN, if the port meets all other requirements. • 3—allowedActive: The port has been configured to be part of the VLAN, and is allowed to be part of the VLAN, if the port meets all other requirements. However, unlike in the case of allowed ports, this port has a device that is participating in the VLAN associated with the port. • 4—allowedNotAvail: The port is active on some other VLAN, and is not available currently. This value applies to devices that do not allow a port to be part of more than one VLAN at the same time. • 5—notAssociated: The port is part of a port group that is not associated with the VLAN. <p>Default value for this object is allowed.</p>
jnxExVlanPortTagness	jnxExVlanPortGroupEntry 4	Specifies whether the port is tagged or untagged.
jnxExVlanPortAccessMode	jnxExVlanPortGroupEntry 5	Specifies whether the port access mode is Access or Trunk.

MAC List Table

The **jnxVlanMacListTable**, whose object identifier is **{jnxVlanMIBObjects 4}**, contains information about MAC address lists. Each **jnxVlanMacListEntry** contains the objects listed in [Table 372 on page 846](#).

Table 372: jnxVlanMacListTable

Object	Object ID	Description
jnxVlanMacListIndex	jnxVlanMacListEntry 1	Uniquely identifies a MAC address list.
jnxVlanMacAddress	jnxVlanMacListEntry 2	Specifies a MAC address that belongs to the group.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 81

VPLS MIB

- [Interpreting the Enterprise-Specific VPLS MIBs on page 849](#)

Interpreting the Enterprise-Specific VPLS MIBs

- [Juniper Networks Enterprise-Specific Implementation of VPLS MIBs on page 849](#)
- [Juniper Networks Enterprise-Specific VPLS Generic MIB on page 850](#)
- [Juniper Networks Enterprise-Specific VPLS LDP MIB on page 854](#)
- [Juniper Networks Enterprise-Specific VPLS BGP MIB on page 855](#)

Juniper Networks Enterprise-Specific Implementation of VPLS MIBs

Starting with Release 11.4, Junos OS extends SNMP support to virtual private LAN service (VPLS) networks, and thus enables users to access VPLS-related data over SNMP.

The Junos OS SNMP support for VPLS is based on the enterprise-specific extension of the following IETF standard MIBs defined in the *Virtual Private LAN Service (VPLS) Management Information Base* Internet draft `draft-ietf-l2vpn-vpls-mib-05.txt`:

- **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`

The enterprise-specific extensions are implemented as part of the `jnxExperiment` branch.

For downloadable versions of the enterprise-specific MIBs, see:

- [Junos OS System Basics and Services Command Reference](#)
- [Junos OS System Basics and Services Command Reference](#)
- [Junos OS System Basics and Services Command Reference](#)

For more information about VPLS MIBs, see the following topics:

- [Juniper Networks Enterprise-Specific VPLS Generic MIB on page 850](#)
- [Juniper Networks Enterprise-Specific VPLS LDP MIB on page 854](#)
- [Juniper Networks Enterprise-Specific VPLS BGP MIB on page 855](#)

Juniper Networks Enterprise-Specific VPLS Generic MIB

The enterprise-specific VPLS Generic MIB is a Juniper Networks implementation of the IETF standard **VPLS-Generic-Draft-01-MIB** defined in draft-ietf-l2vpn-vpls-mib-05.txt. The object identifier for the enterprise-specific version implemented under the **jnxExperiment** branch is **{jnxExperiment 8}**.

This MIB contains generic managed objects definitions for VPLS as defined in RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*, and RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*.

For a downloadable version of this MIB, see [Junos OS System Basics and Services Command Reference](#).

For information about the enterprise-specific VPLS Generic MIB objects, see the following topics:

- [jnxVplsConfigTable](#) on page 850
- [jnxVplsStatusTable](#) on page 851
- [jnxVplsPwBindTable](#) on page 852
- [jnxVplsBgpADConfigTable](#) on page 853
- [jnxVplsBgpRteTargetTable](#) on page 854

[jnxVplsConfigTable](#)

The **jnxVplsConfigTable**, whose object identifier is **{jnxVplsObjects 2}**, contains information related to configuring and monitoring VPLS. The **jnxVplsConfigTable** contains **jnxVplsConfigEntry**, whose object identifier is **{jnxVplsConfigTable 1}**.

Each **jnxVplsConfigEntry** is indexed with **jnxVplsConfigIndex** that uniquely identifies a VPLS, and contains the objects listed in [Table 373 on page 850](#).

Table 373: jnxVplsConfigEntry

Object	Object Identifier	Description
jnxVplsConfigIndex	jnxVplsConfigEntry 1	Contains the unique index that identifies a row specific to a VPLS.
jnxVplsConfigName	jnxVplsConfigEntry 2	Contains the name of the VPLS.
jnxVplsConfigDescr	jnxVplsConfigEntry 3	Contains a brief description of the VPLS.

Table 373: jnxVplsConfigEntry (*continued*)

Object	Object Identifier	Description
jnxVplsConfigAdminStatus	jnxVplsConfigEntry 4	Indicates the administrative state of the VPLS. Possible values are 1 (up), 2 (down), and 3 (testing). The testing (3) state indicates that no operational packets can be passed. The default value is 2 (down).
jnxVplsConfigMacLearning	jnxVplsConfigEntry 6	Indicates whether MAC learning is enabled (true) or not (false). The default value is true .
jnxVplsConfigDiscardUnknownDest	jnxVplsConfigEntry 7	Indicates whether frames received with an unknown destination MAC address are processed (false) or discarded (true). The default value is false .
jnxVplsConfigMacAging	jnxVplsConfigEntry 8	Indicates whether MAC aging is enabled (true) or not (false). The default value is true .
jnxVplsConfigRowStatus	jnxVplsConfigEntry 12	Indicates the row status.
jnxVplsConfigMtu	jnxVplsConfigEntry 13	Specifies the MTU of the VPLS. The range is 64 through 1518. The default value is 1518.
jnxVplsConfigVpnId	jnxVplsConfigEntry 14	Contains the IEEE 802-1990 VPN ID of the associated VPLS.
jnxVplsConfigServiceType	jnxVplsConfigEntry 15	Specifies the type of service—1 (vlan) or 2 (ethernet)—emulated by the VPLS. The default value is 1.
jnxVplsConfigStorageType	jnxVplsConfigEntry 1	Indicates the storage type for the row. The default value is volatile .

jnxVplsStatusTable

The **jnxVplsStatusTable**, whose object identifier is **{jnxVplsObjects 3}**, contains the **jnxVplsStatusEntry**. The **jnxVplsStatusEntry** represents a VPLS instance, and is indexed

by `jnxVplsConfigIndex`. A `jnxVplsStatusEntry` is automatically created when a VPLS instance is configured.

Each `jnxVplsStatusEntry`, whose object identifier is `{jnxVplsStatusTable 1}`, contains the objects listed in [Table 374 on page 852](#):

Table 374: jnxVplsStatusEntry

Object	Object Identifier	Description
<code>jnxVplsStatusOperStatus</code>	<code>jnxVplsStatusEntry 1</code>	Shows the current operational state of the VPLS.
<code>jnxVplsStatusPeerCount</code>	<code>jnxVplsStatusEntry 2</code>	Shows the number of peers for the VPLS.

jnxVplsPwBindTable

The `jnxVplsPwBindTable`, whose object identifier is `{jnxVplsObjects 4}`, provides information about associations between a VPLS instance and pseudowires. A VPLS can be associated with more than one pseudowire. Note that pseudowires are defined in the `pwTable`.

The `jnxVplsPwBindTable` contains the `jnxVplsPwBindEntry`, whose object identifier is `{jnxVplsPwBindTable 1}`. Each `jnxVplsPwBindEntry`, indexed with `jnxVplsConfigIndex` and `jnxVplsPwBindIndex`, contains the objects listed in [Table 375 on page 852](#).

Table 375: jnxVplsPwBindEntry

Object	Object Identifier	Description
<code>jnxVplsPwBindConfigType</code>	<code>jnxVplsPwBindEntry 1</code>	Indicates whether the pseudowire binding was created manually or through autodiscovery. Possible values are 1 (manual) and 2 (autodiscovery).
<code>jnxVplsPwBindType</code>	<code>jnxVplsPwBindEntry 2</code>	Indicates whether the pseudowire is of type mesh or spoke . A value of 1 denotes a mesh type pseudowire, whereas a value of 2 denotes spoke type.
<code>jnxVplsPwBindRowStatus</code>	<code>jnxVplsPwBindEntry 3</code>	Indicates the row status. No object in the table can be modified when the row status is active , denoted by 1.
<code>jnxVplsPwBindStorageType</code>	<code>jnxVplsPwBindEntry 4</code>	Indicates the storage type for the row. The default value is volatile .

Table 375: jnxVplsPwBindEntry (*continued*)

Object	Object Identifier	Description
jnxVplsPwBindIndex	jnxVplsPwBindEntry 5	Contains the secondary index for the conceptual row to identify a pseudowire within the pwEntry . This must match with a pwTable entry that represents an already provisioned pseudowire.

jnxVplsBgpADConfigTable

The **jnxVplsBgpADConfigTable**, whose object identifier is **{jnxVplsObjects 5}**, contains information related to BGP autodiscovery parameters for a VPLS.

The **jnxVplsBgpADConfigEntry**, whose object identifier is **{jnxVplsBgpADConfigTable 1}**, is indexed with **jnxVplsConfigIndex**. Each **jnxVplsBgpADConfigEntry** contains the objects listed in [Table 376 on page 853](#).

Table 376: jnxVplsBgpADConfigEntry

Object	Object Identifier	Description
jnxVplsBgpADConfigRouteDistinguisher	jnxVplsBgpADConfigEntry 1	Contains the router distinguisher for the VPLS. For more information on router distinguisher and its use for a VPLS, see RFC 4364 and RFC 4761.
jnxVplsBgpADConfigPrefix	jnxVplsBgpADConfigEntry 2	Indicates whether the advertised prefix is the loopback IP address or a user-defined value. The default value is 0 , denoting that the advertised prefix is the loopback IP address.
jnxVplsBgpADConfigVplsId	jnxVplsBgpADConfigEntry 3	Contains a unique identifier that is applicable for all VSIs in a VPLS.
jnxVplsBgpADConfigRowStatus	jnxVplsBgpADConfigEntry 4	Indicates the row status.

jnxVplsBgpRteTargetTable

The **jnxVplsBgpRteTargetTable**, whose object identifier is **{jnxVplsObjects 6}**, contains the list of route targets imported or exported by BGP during the autodiscovery of the VPLS.

The **jnxVplsBgpRteTargetEntry**, whose object identifier is **{jnxVplsBgpRteTargetTable 1}**, is indexed with **jnxVplsConfigIndex** and **jnxVplsBgpRteTargetIndex**. Each **jnxVplsBgpRteTargetEntry** contains the objects listed in [Table 377 on page 854](#).

Table 377: jnxVplsBgpRteTargetEntry

Object	Object Identifier	Description
jnxVplsBgpRteTargetIndex	jnxVplsBgpRteTargetEntry 1	In combination with the jnxVplsConfigIndex , this object uniquely identifies an entry in the jnxVplsBgpRteTargetTable . Users can configure multiple route targets for the same VPLS by using new values for this object while keeping the jnxVplsConfigIndex constant.
jnxVplsBgpRteTargetRTType	jnxVplsBgpRteTargetEntry 2	Indicates whether the route target is imported, exported, or both.
jnxVplsBgpRteTargetRT	jnxVplsBgpRteTargetEntry 3	Contains the route target associated with the VPLS.
jnxVplsBgpRteTargetRTRowStatus	jnxVplsBgpRteTargetEntry 4	Indicates the row status.

Juniper Networks Enterprise-Specific VPLS LDP MIB

The enterprise-specific VPLS LDP MIB, **jnx-vpls-ldp-mib**, is a Juniper Networks implementation of the IETF standard VPLS LDP MIB. The enterprise-specific VPLS LDP MIB, whose object identifier is **{jnxExperiment 9}**, contains managed objects definitions for LDP-signaled VPLS (RFC 4762). This MIB also supports data related to pseudowires in the VPLS.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-ldp.txt.

The following topics contain information about VPLS LDP objects supported by Junos OS:

- [jnxVplsLdpConfigTable on page 855](#)
- [jnxVplsLdpPwBindTable on page 855](#)

jnxVplsLdpConfigTable

The **jnxVplsLdpConfigTable**, whose object identifier is **{jnxVplsLdpObjects 1}** contains the **jnxVplsLdpConfigEntry**. Each **jnxVplsLdpConfigEntry** is indexed by the **jnxVplsConfigIndex**, and contains the **jnxVplsLdpConfigMacAddrWithdraw** object. The object identifier for **jnxVplsLdpConfigEntry** is **{jnxVplsLdpConfigTable 1}**.

The **jnxVplsLdpConfigMacAddrWithdraw** object, whose object identifier is **{jnxVplsLdpConfigEntry 1}**, shows the MAC address withdrawal learning status for the VPLS. The value of this object is set to **true** when MAC address withdrawal learning is enabled for the VPLS. The value is set to **false** when MAC address withdrawal learning is disabled.

jnxVplsLdpPwBindTable

The **jnxVplsLdpPwBindTable**, whose object identifier is **{jnxVplsLdpObjects 2}**, contains the **jnxVplsLdpPwBindEntry**. Each **jnxVplsLdpPwBindEntry** is indexed with **jnxVplsConfigIndex** and **jnxVplsPwBindIndex**, and contains the **jnxVplsLdpPwBindMacAddressLimit** object.

The **jnxVplsLdpPwBindMacAddressLimit** object, whose object identifier is **{jnxVplsLdpPwBindEntry 1}**, shows the maximum number of learned and static MAC address entries allowed in the forwarding database for the pseudowire binding. The range is **0** through **4294967295**. The default value is **0**, which denotes that there is no maximum limit for the entries.

Juniper Networks Enterprise-Specific VPLS BGP MIB

The enterprise-specific VPLS BGP MIB is a Juniper Networks implementation of the IETF standard BGP VPLS MIB. The enterprise-specific VPLS BGP MIB, whose object identifier is **{jnxExperiment 10}**, contains managed object definitions for the BGP-signaled VPLS (RFC 4761). This MIB also supports data related to pseudowires in the VPLS.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpls-bgp.txt.

The following topics contain information about VPLS BGP objects supported by Junos OS:

- [jnxVplsBgpConfigTable](#) on page 855
- [jnxVplsBgpVETable](#) on page 856
- [jnxVplsBgpPwBindTable](#) on page 856

jnxVplsBgpConfigTable

The **jnxVplsBgpConfigTable**, whose object identifier is **{jnxVplsBgpObjects 1}**, contains the **jnxVplsBgpConfigEntry**. Each **jnxVplsBgpConfigEntry** is indexed with **jnxVplsConfigIndex**, and contains the **jnxVplsBgpConfigVERangeSize** object. The object identifier for **jnxVplsBgpConfigEntry** is **{jnxVplsBgpConfigTable 1}**.

The **jnxVplsBgpConfigVERangeSize** object, whose object identifier is **{jnxVplsBgpConfigEntry 1}**, specifies the range of the VPLS edge device (VE) identifiers in the VPLS. This number controls the size of the label block advertised for the VE by the provider edge (PE) device. The available range is **0** through **65535**. A value of **0** indicates that the range is not configured, and the PE derives the range value from received advertisements from other PE devices.

jnxVplsBgpVETable

The **jnxVplsBgpVETable**, whose object identifier is **{jnxVplsBgpObjects 2}**, contains information about the association of VE devices with a VPLS. This table contains an entry, **jnxVplsBgpVEEntry**, for each VE device identifier configured on a PE device for a particular VPLS instance.

Each **jnxVplsBgpVEEntry**, whose object identifier is **{jnxVplsBgpVETable 1}**, contains the objects listed in [Table 378 on page 856](#).

Table 378: jnxVplsBgpVEEntry

Object	Object Identifier	Description
jnxVplsBgpVEId	jnxVplsBgpVEEntry 1	Contains a secondary index that identifies a VE within a VPLS instance.
jnxVplsBgpVEName	jnxVplsBgpVEEntry 2	Contains the name of the site or the u-PE associated with the VE ID.
jnxVplsBgpVEPreference	jnxVplsBgpVEEntry 3	Specifies the preference for the VE ID on the PE device if the site is multihomed and the VE ID is reused.
jnxVplsBgpVERowStatus	jnxVplsBgpVEEntry 5	Indicates the status of the row. When this object is set to active , objects in this row cannot be modified.
jnxVplsBgpVEStorageType	jnxVplsBgpVEEntry 6	Indicates the storage type for this row.

jnxVplsBgpPwBindTable

The **jnxVplsBgpPwBindTable**, whose object identifier is **{jnxVplsBgpObjects 3}**, provides BGP-specific information about the association between a VPLS and the corresponding pseudowire. One VPLS can be associated with more than one pseudowire.

Each **jnxVplsBgpPwBindEntry**, whose object identifier is **{jnxVplsBgpPwBindTable 1}**, is indexed with **jnxVplsConfigIndex** and **jnxVplsPwBindIndex**, and contains the objects listed in [Table 379 on page 856](#):

Table 379: jnxVplsBgpPwBindEntry

Object Name	Object Identifier	Description
jnxVplsBgpPwBindLocalVEId	jnxVplsBgpPwBindEntry 1	Identifies the local VE that the pseudowire is associated with.

Table 379: jnxVplsBgpPwBindEntry (continued)

Object Name	Object Identifier	Description
jnxVplsBgpPwBindRemoteVEId	jnxVplsBgpPwBindEntry 2	Identifies the remote VE that the pseudowire is associated with

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 82

VPN Certificate Objects MIB

- [Interpreting the Enterprise-Specific VPN Certificate Objects MIB on page 859](#)

Interpreting the Enterprise-Specific VPN Certificate Objects MIB

- [VPN Certificate Objects MIB on page 859](#)
- [jnxJsLoadedCaCertTable on page 859](#)
- [jnxJsLoadedLocalCertTable on page 860](#)

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/junos13.2/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 859](#)
- [jnxJsLoadedLocalCertTable on page 860](#)

jnxJsLoadedCaCertTable

The **jnxJsLoadedCaCertTable**, whose object ID is **{jnxJsCertificateObjects 1}**, exposes the loaded Certification Authority (CA)-issued certificate objects listed in [Table 380 on page 860](#). Certificates are used when establishing a secure connection in the device.

Table 380: jnxJsLoadedCaCertTable

Object	Object ID	Description
jnxJsLoadedCaCertEntry	jnxJsLoadedCaCertTable 1	<p>Loaded CA certificate entry. The loaded CA certificate entry is indexed by the CA certification name.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • jnxJsLoadedCaCertName • jnxJsLoadedCaCertSubject • jnxJsLoadedCaCertExpire • jnxJsLoadedCaCertIssuer
jnxJsLoadedCaCertName	jnxJsLoadedCaCertEntry 1	Loaded CA certificate name. This name is unique within the loaded CA certificates.
jnxJsLoadedCaCertSubject	jnxJsLoadedCaCertEntry 2	CA certificate subject.
jnxJsLoadedCaCertExpire	jnxJsLoadedCaCertEntry 3	Expiration time and date of CA certificate.
jnxJsLoadedCaCertIssuer	jnxJsLoadedCaCertEntry 4	Issuer of CA certificate.

jnxJsLoadedLocalCertTable

The **jnxJsLoadedLocalCertTable**, whose object ID is **{jnxJsCertificateObjects 2}**, exposes the loaded local certificate objects listed in [Table 381 on page 860](#). Certificates are used when establishing a secure connection in the device.

Table 381: jnxJsLoadedLocalCertTable

Object	Object ID	Description
jnxJsLoadedLocalCertEntry	jnxJsLoadedLocalCertTable 1	<p>Default certificate entry. This entry is indexed by the certification name.</p> <p>Sequence of parameters:</p> <ul style="list-style-type: none"> • 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.
jnxJsLoadedLocalCertIssuer	jnxJsLoadedLocalCertEntry 4	Issuer of the local certificate.

**Related
Documentation**

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

CHAPTER 83

VPN MIB

- Interpreting the Enterprise-Specific VPN MIB on page 863

Interpreting the Enterprise-Specific VPN MIB

- VPN MIB on page 863
- jnxVpnInfo on page 864
- jnxVpnTable on page 864
- jnxVpnIfTable on page 865
- jnxVpnPwTable on page 868
- jnxVpnRTTable on page 873
- VPN Traps on page 873

VPN MIB

The Juniper Networks enterprise-specific Virtual Private Network (VPN) MIB, whose object identifier is **{jnxMibs 26}**, provides monitoring for the following type of VPNs:

- Layer 2 based on Internet draft draft-kompella-l2ppvpn-version.txt, *MPLS-based Layer 2 VPNs*.
- Layer 3 based on Internet draft draft-ietf-l3vpn-rfc2547bis-03.txt, *BGP and MPLS IP VPNs*.
- VPLS based on Internet draft draft-ietf-ppvpn-vpls-bgp-00.txt, *Virtual Private LAN Service*.



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

For a downloadable version of this MIB, see

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vpn.txt.

For information about the enterprise-specific VPN MIB objects, see the following topics:

- [jnxVpnInfo on page 864](#)
- [jnxVpnTable on page 864](#)
- [jnxVpnIfTable on page 865](#)
- [jnxVpnPwTable on page 868](#)
- [jnxVpnRTTable on page 873](#)
- [VPN Traps on page 873](#)

jnxVpnInfo

jnxVpnInfo, whose object identifier is **{jnxVpnMibObjects 1}**, contains information about the number of configured VPNs and active VPNs.

[Table 382 on page 864](#) lists the supported **jnxVpnInfo** objects, VPNs, and circuit connection services.

Table 382: Supported jnxVpnInfo Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	Circuit Cross-Connect	L2 Circuit	Optical VPN	Description
jnxVpnConfiguredVpns	jnxVpnInfo 1	Yes	Yes	Yes	No	Yes	—	Number of configured VPNs.
jnxVpnActiveVpns	jnxVpnInfo 2	Yes	Yes	Yes	No	Yes	—	Number of active VPNs.
jnxVpnNextIfIndex	jnxVpnInfo 3	—	—	—	—	—	—	Next free VPN interface index.
jnxVpnNextPwIndex	jnxVpnInfo 4	—	—	—	—	—	—	Next free pseudowire index.
jnxVpnNextRTIndex	jnxVpnInfo 5	—	—	—	—	—	—	Next free route target index.

jnxVpnTable

jnxVpnTable, whose object identifier is **{jnxVpnMibobjects 2}**, lists configured VPNs.

jnxVpnEntry contains information about a configured VPN with the objects listed in [Table 383 on page 865](#) and their supported VPNs and circuit connection services. The first two objects in **jnxVpnEntry** (**jnxVpnType** and **jnxVpnname**) are indexes and are not included in this table.

Table 383: Supported `jnxVpnEntry` Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
<code>jnxVpnRowStatus</code>	<code>jnxVpnEntry 3</code>	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
<code>jnxVpnStorageType</code>	<code>jnxVpnEntry 4</code>	–	–	–	–	–	–	The storage type.
<code>jnxVpnDescription</code>	<code>jnxVpnEntry 5</code>	Yes	Yes	Yes	Yes	No	–	VPN description.
<code>jnxVpnIdentifierType</code>	<code>jnxVpnEntry 6</code>	Yes	Yes	Yes	Yes	No	–	Type of <code>jnxVpnIdentifier</code> .
<code>jnxVpnIdentifier</code>	<code>jnxVpnEntry 7</code>	Yes	Yes	Yes	Yes	No	–	For BGP VPNs, the route distinguisher for the VPN. For LDP VPNs, the virtual circuit (VC) ID for the circuit. A value of all zeros indicates that a route distinguisher and a VC ID are not configured for the VPN.
<code>jnxVpnConfiguredSites</code>	<code>jnxVpnEntry 8</code>	–	No	No	No	No	–	The number of sites configured in the VPN.
<code>jnxVpnActiveSites</code>	<code>jnxVpnEntry 9</code>	–	No	No	No	No	–	The number of active sites in the VPN.
<code>jnxVpnLocalAddresses</code>	<code>jnxVpnEntry 10</code>	No	No	No	No	No	–	The number of addresses learned from the CE device.
<code>jnxVpnTotalAddresses</code>	<code>jnxVpnEntry 11</code>	No	No	No	No	No	–	The total number of addresses in the VPN routing table.
<code>jnxVpnVpnAge</code>	<code>jnxVpnEntry 12</code>	Yes	Yes	Yes	Yes	No	–	The age of the VPN, in hundredths of a second.

`jnxVpnIfTable`

The `jnxVpnIfTable`, whose object identifier is `{jnxVpnMibObjects 3}`, lists VPN interfaces.

jnxVpnIfEntry contains information about VPN interfaces, and has the objects listed in [Table 384 on page 866](#). The first three objects (**jnxVpnIfVpnType**, **jnxVpnIfVpnName**, and **jnxVpnIfIndex**) are indexes and are not included in this table.

Table 384: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfRowStatus	jnxVpnIfEntry 4	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
jnxVpnIfStorageType	jnxVpnIfEntry 5	–	–	–	–	–	–	Identifies the storage type for an object.
jnxVpnIfAssociationPw	jnxVpnIfEntry 6	–	Yes	Yes	Yes	No	–	The index of the associated pseudowire. If no index is associated with a pseudowire, the index is 0. A pseudowire is a mechanism that carries essential elements of an emulated circuit from one provider edge (PE) device to one or more other PEs over a PSN.

Table 384: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfProtocol	jnxVpnIfEntry 7	No	Yes	Yes	Yes	No	–	<p>Indicates the protocol running over a VPN interface.</p> <p>This object contains the following values:</p> <ul style="list-style-type: none"> • other(0) • frameRelay(1) • atmAal5(2) • atmCell(3) • ethernetVlan(4) • ethernet(5) • ciscoHdlc(6) • ppp(7) • cem(8) • atmVcc(9) • atmVpc(10) • vpls(11) • ipInter-working(12) • snapInter-working(13) • static(20) • rip(21) • ospf(22) • bgp(23) • atmTrunkNNI (129) • atmTrunkUNI (130)
jnxVpnIfInBandwidth	jnxVpnIfEntry 8	No	No	No	No	No	–	<p>The maximum bandwidth that the customer edge (CE) device connected over a VPN can send to the PE device, in Kbps. A value of 0 indicates that there is no configured maximum.</p>

Table 384: Supported jnxVpnIfEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfOutBandwidth	jnxVpnIfEntry 9	No	No	No	No	No	—	The maximum bandwidth that the PE device can send to the CE device over a VPN interface, in Kbps. A value of 0 indicates that there is no configured maximum.
jnxVpnIfStatus	jnxVpnIfEntry 10	Yes	Yes	Yes	Yes	No	—	<p>Status of a monitored VPN interface.</p> <p>This object contains the following values:</p> <ul style="list-style-type: none"> • unknown(0) • noLocal-Interface(1) • disabled(2) • encapsulation-Mismatch(3) • down(4) • up(5)

jnxVpnPwTable

jnxVpnPwTable, whose object identifier is **{jnxVpnMibObjects 4}**, lists pseudowire connections.

jnxVpnPwEntry contains pseudowire information about a VPN that is being monitored, and has the objects listed in [Table 385 on page 868](#). The first three objects (**jnxVpnPwVpnType**, **jnxVpnPwVpnName**, and **jnxVpnPwIndex**) are indexes and are not listed in this table.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwRowStatus	jnxVpnPwEntry 4	—	—	—	—	—	—	Creates, modifies, and deletes a row in this table.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwStorageType	jnxVpnPwEntry 5	–	–	–	–	–	–	The storage type.
jnxVpnPwAssociatedInterface	jnxVpnPwEntry 6	–	Yes	Yes	Yes	No	–	The VPN index of the interface associated with a pseudowire. If no interface is associated with a pseudowire, 0 is returned.
jnxVpnPwLocalSiteId	jnxVpnPwEntry 7	–	Yes	Yes	Yes	No	–	The local site identifier for a pseudowire. When there is no local site identifier, 0 is returned.
jnxVpnPwRemoteSiteId	jnxVpnPwEntry 8	–	Yes	Yes	Yes	No	–	The remote site identifier. For example, the site at the end of the pseudowire. When there is no remote site identifier, 0 is returned.
jnxVpnRemotetPeldAddrType	jnxVpnPwEntry 9	–	Yes	Yes	Yes	No	–	The remote PE address. For example, the router at the end of the pseudowire.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRemotePeldAddress	jnxVpnPwEntry 10	—	Yes	Yes	Yes	No	—	<p>The type of tunnel over which the pseudowire is carried. If several pseudowires can be carried in one tunnel, each pseudowire is identified by the multiplexer or demultiplexer within a tunnel.</p> <p>This object can contain the following values:</p> <ul style="list-style-type: none"> • static(1) • gre(2) • l2tpv3(3) • ipSec(4) • ldp(5) • rsvpTe(6) • crLdp(7)
jnxVpnPwTunnelType	jnxVpnPwEntry 11	—	Yes	Yes	Yes	No	—	The type of tunnel over which the pseudowire is carried.
jnxVpnPwTunnelName	jnxVpnPwEntry 12	—	Yes	Yes	Yes	No	—	The name of the tunnel over which a pseudowire is carried.
jnxVpnPwReceiveDemux	jnxVpnPwEntry 13	—	Yes	Yes	Yes	No	—	The demultiplexer value that identifies received packets associated with this pseudowire.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwTransmitDemux	jnxVpnPwEntry 14	–	Yes	Yes	Yes	No	–	The demultiplexer value that identifies the transmitted packets associated with this pseudowire.
jnxVpnPwStatus	jnxVpnPwEntry 15	–	Yes	Yes	Yes	No	–	<p>The status of the pseudowire.</p> <p>This object can have the following values:</p> <ul style="list-style-type: none"> • unknown(0) • down(1) • up(2)
jnxVpnPwTunnelStatus	jnxVpnPwEntry 16	–	No	No	No	No	–	The status of the PE-to-PE tunnel over which the pseudowire is carried.
jnxVpnPwRemoteSiteStatus	jnxVpnPwEntry 17	–	No	No	No	No	–	<p>The interface status at the remote end of the pseudowire.</p> <p>This object can have the following values:</p> <ul style="list-style-type: none"> • unknown(0) • outOf-Range(1) • down(2) • up(3)
jnxVpnPwTimeUp	jnxVpnPwEntry 18	–	Yes	Yes	Yes	No	–	The time, in hundredths of a second, that a pseudowire has been operational.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwTransitions	jnxVpnPwEntry 19	–	Yes	Yes	Yes	No	–	The number of state transitions (up to down and down to up) that a tunnel has undergone.
jnxVpnPwLastTransition	jnxVpnPwEntry 20	–	Yes	Yes	Yes	No	–	The time, in hundredths of a second, since the last transition occurred in a tunnel.
jnxVpnPwPacketsSent	jnxVpnPwEntry 21	–	No	No	No	No	–	The number of packets sent over a pseudowire.
jnxVpnPwOctetsSent	jnxVpnPwEntry 22	–	No	No	No	No	–	The number of octets sent over a pseudowire.
jnxVpnPwPacketsReceived	jnxVpnPwEntry 23	No	No		No	No	–	The number of packets received over a pseudowire.
jnxVpnPwOctetsReceived	jnxVpnPwEntry 24	No	No		No	No	–	The number of octets received over a pseudowire.
jnxVpnPwLRPacketsSent	jnxVpnPwEntry 25	No	No		No	No	–	The number of packets sent over a pseudowire.
jnxVpnPwLROctetsSent	jnxVpnPwEntry 26	No	No		No	No	–	The number of octets sent over a pseudowire.
jnxVpnPwLRPacketsReceived	jnxVpnPwEntry 27	No	No		No	No	–	The number of packets received over a pseudowire.

Table 385: Supported jnxVpnPwEntry Objects, VPNs, and Circuit Connection Services (*continued*)

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnPwLROctetsReceived	jnxVpnPwEntry 28	No	No		No	No	–	The number of octets received over a pseudowire.

jnxVpnRTTable

The **jnxVpnRTTable**, whose object identifier is **{jnxVpnMibObjects 4}**, contains route targets for a VPN.

jnxVpnRTEntry lists route targets for a given VPN, and has the objects listed in [Table 386 on page 873](#). The first three objects (**jnxVpnRTVpnType**, **jnxVpnRTVpnName**, and **jnxVpnRTIndex**) are indexes and are not listed in this table.

Table 386: Supported jnxVpnRTEntry Objects, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnRTRowStatus	jnxVpnRTEntry 4	–	–	–	–	–	–	Creates, modifies, or deletes a row in this table.
jnxVpnRTStorageType	jnxVpnRTEntry 5	–	–	–	–	–	–	Identifies the storage type for an object.
jnxVpnRTType	jnxVpnRTEntry 6	Yes	Yes	Yes	–	No	–	The type of the following route target. The type can be routeTarget[012] or none .
jnxVpnRT	jnxVpnRTEntry 7	Yes	Yes	Yes	–	No	–	The VPN route target. If jnxVpnRTType is none , the value must be all zeros.
jnxVpnRTFunction	jnxVpnRTEntry 8	Yes	Yes	Yes	–	No	–	The route target export distribution type.

VPN Traps

The enterprise-specific VPN MIB provides traps for monitoring VPNs. [Table 387 on page 874](#) lists supported VPN traps, VPNs, and circuit connection services.

Table 387: Supported VPN Traps, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfUp	jnxVpnMIBnotificatio 1	Yes	Yes	–	Yes	No	–	Indicates that the interface with the index jnxVpnIfIndex belonging to the jnxVpnIfVpnName of type jnxVpnIfVpnType went up.
jnxVpnIfDown	jnxVpnMIBnotificatio 2	Yes	Yes	–	Yes	No	–	Indicates that the interface with index jnxVpnIfIndex belonging to jnxVpnIfVpnName of type jnxVpnIfVpnType went down.
jnxVpnPwUp	jnxVpnMIBnotificatio 3	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with the index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went up.
jnxVpnPwDown	jnxVpnMIBnotificatio 4	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went down.

Related Documentation

- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Juniper Networks Enterprise-Specific MIBs and Supported Devices on page 36](#)
- [SNMP MIB Objects Supported by Junos OS for the Set Operation on page 46](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

PART 3

SNMP Notifications

- [Enterprise-Specific Traps on page 877](#)
- [Standard Traps on page 913](#)

CHAPTER 84

Enterprise-Specific Traps

- Antivirus Traps Objects Table on page 877
- Notifications for the BFD MIB on page 878
- Juniper Networks Enterprise-Specific BGP Traps on page 878
- Chassis Traps on page 879
- Configuration Management Notifications on page 884
- DFC Notification Definitions on page 885
- Notifications for the Event MIB on page 886
- jnxTraps on page 886
- Juniper Networks Enterprise-Specific LDP Traps on page 887
- Juniper Networks Enterprise-Specific License MIB Notifications on page 890
- MAC Address Limit Traps on page 890
- Juniper Networks Enterprise-Specific MIMSTP Traps on page 891
- Juniper Networks Enterprise-Specific MPLS Traps on page 892
- Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways on page 892
- Juniper Networks Enterprise-Specific NAT Traps on MX Series 3D Universal Edge Routers on page 893
- Service Traps on page 894
- Juniper Networks Enterprise-Specific SNMP Version 1 Traps on page 894
- Juniper Networks Enterprise-Specific SNMP Version 2 Traps on page 902
- Juniper Networks Enterprise-Specific Traps on EX Series Switches on page 910
- Juniper Networks Enterprise-Specific Traps on MX Series 3D Universal Edge Routers on page 911
- VPN Traps on page 911

Antivirus Traps Objects Table

The **jnxJsAvPatternUpdateTrap** object, whose object ID is **{jnxJsAntiVirusTrapsPrefix 1}**, generates a trap when the antivirus pattern database is updated.

Table 22 on page 87 describes the objects in the Antivirus Traps Objects Table.

Table 388: Antivirus Traps Objects Table

Object	Object ID	Description
<code>jnxAVPatternVersionString</code>	<code>jnxJsAntiVirusTrapVars 1</code>	The last successful update of the antivirus pattern database version.
<code>jnxAVPatternTimestamp</code>	<code>jnxJsAntiVirusTrapVars 2</code>	The last successful update of the antivirus pattern database timestamp.

- Related Documentation**
- [Antivirus Objects MIB on page 85](#)
 - [Scan Engine Objects Table on page 86](#)
 - [Scan Statistics Objects Table on page 86](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

Notifications for the BFD MIB

Table 32 on page 108 lists the supported notifications for the BFD MIB.

Table 389: Supported Notifications for the BFD MIB

Object	Object Identifier	Description
<code>jnxBfdSessTxIntervalHigh</code>	<code>jnxBfdNotification 1</code>	A notification generated when the threshold value for the transmit interval is configured (<code>jnxBfdSessThresTxInterval</code>) and the BFD session transmit interval (<code>jnxBfdSessCurrTxInterval</code>) 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.
<code>jnxBfdSessDetectionTimeHigh</code>	<code>jnxBfdNotification 2</code>	A notification generated when the threshold value for the detection time is configured (<code>jnxBfdSessThresDectTime</code>) and the BFD session detection time (<code>jnxBfdSessCurrDectTime</code>) 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 107](#)
 - [jnxBfdSessTable on page 107](#)

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 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 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

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 57](#).

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 a six-input DC power supply, you must also inspect the **feeds** configuration. For the six-input DC power supply, the **jnxPowerSupplyFailure** trap is sent when the number of input feeds connected to the power supply is less than the configured value.



NOTE:

1. The number of input feeds can be configured by using the **set chassis pem feed number-of-input-feeds** statement at the **[edit chassis pem]** or **[edit chassis lcc lcc-number pem]** hierarchy level.
2. The behavior stated here for the six-input DC power supply is applicable only on T640, T6100, and T4000 platforms.
3. For a two-input DC power supply, no SNMP trap is generated when either feed fails.

- **Fan failure (jnxFanFailure)**—Sent when the fan fuse blows or when the fan wiring shorts out. When only one fan has failed, there is no service impact. The remaining fans increase speed to compensate. However, you must resolve the problem before another fan fails. This trap is repeated every hour until the fan failure is fixed. To determine the

source of the failure, you must physically inspect the router, taking care to check the fuses. See the hardware installation guide for your router model for more information.

- **Overtemperature (jnxOverTemperature)**—Sent when several fans fail or the room temperature increases significantly. The service impact of this trap depends on the temperature of the router. In general, the router increases the speed of the fans when any component exceeds a temperature of 55° C. The fans remain at the higher speed until the temperature decreases below the threshold. In this case, there is no service impact. However, if the temperature exceeds 75° C, the router transmits a warning and automatically shuts down. This scenario creates a significant service impact because the shutdown affects additional routers and equipment. This trap is repeated every minute until the temperature is brought down to normal. To determine the source of the overtemperature problem, you must physically inspect the router to determine whether any fans have failed in the router.
- **Power Supply OK (jnxPowerSupplyOK)**—Sent when a power supply recovers from failure. (The six-input DC power supply fails when the number of input feeds connected to the power supply is less than the configured value.)

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.
- **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 881](#)
- [SNMPv2 Trap Format on page 882](#)

SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in [Table 92 on page 349](#). To view the SNMPv1 chassis-related traps, see “[Juniper Networks Enterprise-Specific SNMP Version 1 Traps](#)” on page 894.

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 390: SNMP Version 1 Trap Format

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2
jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2
jnxFruCheck	1.3.6.1.4.1.2636.4.1	6	12
jnxFruFailed	1.3.6.1.4.1.2636.4.1	6	9
jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6
jnxFruOffline	1.3.6.1.4.1.2636.4.1	6	10
jnxFruOnline	1.3.6.1.4.1.2636.4.1	6	11
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8
jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5
jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2	6	1
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1	6	4
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3

SNMPv2 Trap Format

The SNMPv2 trap format for the Chassis MIB traps is described in [Table 93 on page 351](#).

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 391: 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.
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.

Table 391: SNMP Version 2 Trap Format (*continued*)

Trap Name	snmpTrapOID	Description
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 on page 116](#)
- [Chassis MIB Textual Conventions on page 117](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

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 375](#)
- [Configuration Management MIB Textual Conventions on page 375](#)
- [Configuration Change Management Objects and jnxCmCfgChgEventTable on page 376](#)
- [Rescue Configuration Change Management Objects on page 378](#)

DFC Notification Definitions

Table 120 on page 419 lists the supported notification definitions.

Table 392: Supported Notification Definitions for the DFC MIB

Notification Type	Objects	Identifier	Description
jnxDfcSoftPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 1	Notification that occurs when the input packet rate (in pps) exceeds the configured limit.
jnxDfcSoftPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsSoftOverloadLowWatermark jnxDfcPpsSoftOverloadHighWatermark	jnxDfcNotificationPrefix 2	Notification that occurs when the input packet rate (in pps) returns to below the configured limit.
jnxDfcHardPpsThresholdExceeded	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 3	Notification that occurs when the input packet rate (in pps) exceeds the recommended limit.
jnxDfcHardPpsUnderThreshold	jnxDfcInputPktRate jnxDfcPpsHardOverloadLowWatermark jnxDfcPpsHardOverloadHighWatermark	jnxDfcNotificationPrefix 4	Notification that occurs when the input packet rate (in pps) returns to below the recommended limit.
jnxDfcSoftMemThresholdExceeded	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 5	Notification that occurs when memory usage exceeds the configured limit.
jnxDfcSoftMemUnderThreshold	jnxDfcFlowUsage jnxDfcCriteriaUsage jnxDfcMemSoftOverloadLowWatermark jnxDfcMemSoftOverloadHighWatermark	jnxDfcNotificationPrefix 6	Notification that occurs when memory usage returns to below the configured limit.
jnxDfcHardMemThresholdExceeded	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 7	Notification that occurs when memory usage exceeds the recommended limit.
jnxDfcHardMemUnderThreshold	jnxDfcFlowUsage jnxDfcFlowLowWatermark jnxDfcFlowHighWatermark jnxDfcCriteriaUsage jnxDfcCriteriaLowWatermark jnxDfcCriteriaHighWatermark	jnxDfcNotificationPrefix 8	Notification that occurs when memory usage returns to below the recommended limit.

- Related Documentation**
- [Dynamic Flow Capture MIB on page 413](#)
 - [jnxDfcCSTable on page 414](#)
 - [jnxDfcCDTable on page 417](#)
 - [DFC Notification Variables on page 418](#)

Notifications for the Event MIB

Table 123 on page 424 lists the supported notifications for the Event MIB.

Table 393: 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 423](#)
 - [jnxEventAvTable on page 423](#)

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}**.
- **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 819](#)
- [jnxProducts on page 819](#)
- [jnxServices on page 820](#)
- [jnxMibs on page 821](#)
- [jnxExperiment on page 425](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific LDP Traps

For information about the enterprise-specific LDP traps, see “LDP MIB” on page 557.

LDP Notification Objects and Notification Types

The **jnxLdpTrapVars** table, whose object identifier is **{jnxLdp 1}**, defines the objects used in the enterprise-specific LDP traps. For more information about LDP notification types, see [Table 178 on page 558](#).



NOTE: The enterprise-specific LDP MIB requires **jnxLdpTrapPrefix** with 0 subidentifier for seamless translation of SNMPv2 notifications to SNMPv1 format.

Table 394: LDP Notification Objects

Object	Object ID	Description
jnxLdpLspFec	jnxLdpTrapVars 1	Contains the LSP forwarding equivalence class (FEC) in IpAddress format.
jnxLdpRtrid	jnxLdpTrapVars 2	Contains the router ID of the sending router in IpAddress format.
jnxLdpLspDownReason	jnxLdpTrapVars 3	<p>Returns one of the following integer values to denote what might have caused the LSP to go down:</p> <ul style="list-style-type: none"> • 1—Change in topology • 2—Withdrawal of label by the neighbor • 3—Unavailability of the neighbor because the neighbor is down • 4—Change in filter • 5—Unknown reason

Table 394: LDP Notification Objects (*continued*)

Object	Object ID	Description
jnxLdpSesDownReason	jnxLdpTrapVars 4	<p>Returns one of the following integer values to denote what might have caused the session transition into nonexistent state:</p> <ul style="list-style-type: none"> • 0—Unknown reason • 1—Hold time expired • 2—Connection time expired • 3—All adjacencies are down • 4—Received bad TLV (type, length, and value encoding scheme) • 5—Received bad PDU (protocol data unit) • 6—Connection error occurred • 7—The connection was reset • 8—Received notification from peer • 9—Received unexpected end-of-file message • 10—The authentication key was changed • 11—Error occurred during initialization • 12—Graceful restart was aborted • 13— CLI command was issued to end the session
jnxLdpSesDownIf	jnxLdpTrapVars 5	Contains the SNMP index of the interface associated with the session-down event. If no interface is associated with the session-down event, then this object returns the SNMP index of any interface associated with one of the neighbors.
jnxLdpLspFecLen	jnxLdpTrapVars 6	Represents the length of the LSP FEC prefix in bits. The allowable range is 0 through 32 bits.
jnxLdpSesUpIf	jnxLdpTrapVars 7	Contains the interface index of one of the neighbors associated with the session.
jnxLdpInstanceName	jnxLdpTrapVars 8	Contains the name of the VPN interface.

[Table 179 on page 559](#) lists the enterprise-specific LDP notifications and the objects contained in each notification. The enterprise-specific LDP notifications use the objects listed in [Table 178 on page 558](#).

Table 395: LDP Notification Types

Object	Object ID	Description
jnxLdpLspUp	jnxLdpTrapPrefix 1	<p>Generated when an LSP comes back online. Typically, this trap is generated only when an LSP that has an active jnxLdpLspDown trap comes back online.</p> <ul style="list-style-type: none"> • jnxLdpLspFec • jnxLdpRtrid • jnxLdpLspFecLen • jnxLdpInstanceName

Table 395: LDP Notification Types (*continued*)

Object	Object ID	Description
<code>jnxLdpLspDown</code>	<code>jnxLdpTrapPrefix 2</code>	<p>Generated when an LSP goes offline. This trap contains the following objects:</p> <ul style="list-style-type: none"> <code>jnxLdpLspFec</code> <code>jnxLdpRtrid</code> <code>jnxLdpLspDownReason</code> <code>jnxLdpLspFecLen</code> <code>jnxLdpInstanceName</code> <p>NOTE: For every <code>jnxLdpLspDown</code> trap generated, Junos OS ensures that a <code>jnxLdpLspUp</code> trap is generated when the LSP comes back online.</p>
<code>jnxLdpSesUp</code>	<code>jnxLdpTrapPrefix 3</code>	<p>Generated when the <code>jnxMplsLdpSesState</code> object moves into the operational (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> <code>jnxMplsLdpSesState</code> <code>jnxLdpSesUpIf</code>
<code>jnxLdpSesDown</code>	<code>jnxLdpTrapPrefix 4</code>	<p>Generated when the <code>jnxMplsLdpSesState</code> object moves out of the operational (5) state. This trap contains the following objects:</p> <ul style="list-style-type: none"> <code>jnxMplsLdpSesState</code> <code>jnxLdpSesDownReason</code> <code>jnxLdpSesDownIf</code> <p>The <code>jnxLdpSesDownIf</code> object contains the address of the interface associated with the last neighbor when the value of <code>jnxLdpSesDownReason</code> was <code>allAdjacenciesDown (3)</code>.</p>

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 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific License MIB Notifications

Table 183 on page 565 lists the license MIB notifications.

Table 396: License MIB Notifications

Notification	Object Identifier	Description
<code>jnxLicenseGraceExpired</code>	<code>jnxLicenseNotifications 1</code>	Generated when the grace period for renewing the license for the feature (identified by <code>jnxLicenseFeatureName</code>) has expired.
<code>jnxLicenseGraceAboutToExpire</code>	<code>jnxLicenseNotifications 2</code>	Generated when the grace period for renewing the license for the feature (identified by <code>jnxLicenseFeatureName</code>) is about to expire.
<code>jnxLicenseAboutToExpire</code>	<code>jnxLicenseNotifications 3</code>	Generated when the license for the feature (identified by <code>jnxLicenseFeatureName</code>) is about to expire.
<code>jnxLicenseInfringeCumulative</code>	<code>jnxLicenseNotifications 4</code>	Generated when the use of the feature (identified by <code>jnxLicenseFeatureName</code>) exceeds the maximum limit specified in the license.
<code>jnxLicenseInfringeSingle</code>	<code>jnxLicenseNotifications 5</code>	Generated when the license for the feature (identified by <code>jnxLicenseFeatureName</code>) is invalid; that is, either expired or not available.

Related Documentation

- [License MIB on page 563](#)
- [jnxLicenseInstallTable on page 564](#)
- [Supported Features and Configuration Parameters for Licensing on page 564](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)

MAC Address Limit Traps

The enterprise-specific L2ALD MIB defines the following traps:

- `jnxL2aldRoutingInstMacLimit`, whose object identifier is `{jnxL2aldNotification 1}`, is generated when the number of MAC addresses for the given routing instance, `jnxL2aldRoutingInst`, exceeds the set limit. This trap contains the following objects: `jnxL2aldLogicalRouter`, `jnxL2aldRoutingInst`, `jnxL2aldBridgeDomain`, and `jnxL2aldMacLimit`.
- `jnxL2aldInterfaceMacLimit`, whose object identifier is `{jnxL2aldNotification 2}`, is generated when the number of MAC addresses for the given physical interface exceeds the set limit. This trap contains the following objects: `jnxL2aldIntfLogicalRouter`, `jnxL2aldIntfRoutingInst`, `jnxL2aldIntfBridgeDomain`, `ifDescr`, and `jnxL2aldIntfMacLimit`.
- `jnxL2aldGlobalMacLimit`, whose object identifier is `{jnxL2aldNotification 3}`, is generated when the MAC limit for the entire system exceeds the set limit.

- Related Documentation**
- [L2ALD MIB on page 537](#)
 - [jnxL2aldInterfaceTable on page 537](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific MIMSTP Traps

Table 241 on page 620 lists the enterprise-specific MIMSTP traps.

Table 397: Enterprise-Specific MIMSTP Traps

Object	Object Identifier	Description
jnxMIMstGenTrap	jnxMIMstTraps 1	Generated when any of the general events such as protocol up or protocol down occurs.
jnxMIMstErrTrap	jnxMIMstTraps 2	Generated when any of the error events such as a memory failure, buffer failure, protocol migration, or new root or topology change occurs.
jnxMIMstNewRootTrap	jnxMIMstTraps 3	Generated when a new root bridge is selected in the topology. The jnxMIMstNewRootTrap indicates that the sending agent has become the new root of the spanning-tree. The trap is sent by a bridge soon after its election as the new root
jnxMIMstTopologyChgTrap	jnxMIMstTraps 4	Generated when a topology change is detected.
jnxMIMstProtocolMigrationTrap	jnxMIMstTraps 5	Generated when a port protocol migration happens on the port.
jnxMIMstInvalidBpduRxdTrap	jnxMIMstTraps 6	Generated when an invalid packet is received for bpdu/stp/rstp/maximum age/forward delay/hello time .
jnxMIMstRegionConfigChangeTrap	jnxMIMstTraps 7	Generated when the multiple spanning-tree region's configuration identifier changes.

- Related Documentation**
- [MIMSTP MIB on page 605](#)
 - [jnxMIDot1sJuniperMstTable on page 606](#)
 - [jnxMIMstMstiBridgeTable on page 612](#)
 - [jnxMIMstVlanInstanceMappingTable on page 614](#)
 - [jnxMIMstCistPortTable on page 615](#)
 - [jnxMIMstMstiPortTable on page 618](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific MPLS Traps

Table 245 on page 630 lists the enterprise-specific MPLS traps based on `mplsLspInfoName`.

Table 398: MPLS Traps

Object	Object ID	Description
<code>mplsLspInfoUp</code>	<code>mplsLspTraps 1</code>	Indicates that the LSP (<code>mplsLspInfoName</code>) is up. The current active path is represented by <code>mplsPathInfoName</code> .
<code>mplsLspInfoDown</code>	<code>mplsLspTraps 2</code>	Indicates that the LSP (<code>mplsLspInfoName</code>) is down because the current active path (<code>mplsPathInfoName</code>) has gone down.
<code>mplsLspInfoChange</code>	<code>mplsLspTraps 3</code>	Indicates that the LSP (<code>mplsLspInfoName</code>) has switched traffic to a new active path (<code>mplsPathInfoName</code>) without changing the state (up) before or after the switch.
<code>mplsLspInfoPathDown</code>	<code>mplsLspTraps 4</code>	Indicates that the specified path (<code>mplsPathInfoName</code>) for the LSP (<code>mplsLspInfoName</code>) has gone down.
<code>mplsLspInfoPathUp</code>	<code>mplsLspTraps 5</code>	Indicates that the specified path (<code>mplsPathInfoName</code>) for the LSP (<code>mplsLspInfoName</code>) has come up.

Related Documentation

- [MPLS MIB on page 625](#)
- [mplsInfo on page 625](#)
- [mplsTEInfo on page 626](#)
- [mplsAdminGroup on page 626](#)
- [mplsLspInfoList on page 627](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific NAT Traps on SRX Series Services Gateways

Table 251 on page 638 lists NAT trap definition objects on SRX Series Services Gateways.

Table 399: NAT Trap Definitions

Object	Object ID	Description
<code>jnxJsNatAddrPoolThresholdStatus</code>	<code>jnxJsNatNotifications 1</code>	<p>NAT address pool utilization threshold status trap to signify that the address pool utilization either exceeds a certain percentage or is clear of that percentage.</p> <p><code>jnxJsNatSrcIpPoolName</code> is the name of the resource pool.</p> <p><code>jnxJsNatAddrPoolUtil</code> is the percentage of utilization of the address pool.</p>

Table 399: NAT Trap Definitions (*continued*)

Object	Object ID	Description
<code>jnxJsNatAddrPoolUtil</code>	<code>jnxJsNatTrapVars 1</code>	Dynamic address pool utilization expressed as a percentage.

**Related
Documentation**

- [NAT Objects MIB on page 633](#)
- [Source NAT Table on page 634](#)
- [New Source NAT Table on page 635](#)
- [jnxJsNatIfSrcPoolPortTable on page 636](#)
- [NAT Rule Hit Table on page 637](#)
- [NAT Pool Hit Table on page 637](#)

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

The `jnxNatAddrPoolThresholdStatus` and `jnxNatAddrPoolUtil` objects notify the user about the usage of NAT resources in percentage.

[Table 255 on page 642](#) lists the NAT trap definition objects.

Table 400: NAT Trap Definitions

Object	Object ID	Description
<code>jnxNatAddrPoolUtil</code>	<code>jnxNatTrapVars 1</code>	Displays the dynamic address pool utilization in percentage.
<code>jnxNatTrapSrcPoolName</code>	<code>jnxNatTrapVars 2</code>	Displays the source NAT pool name that issues the trap.
<code>jnxNatAddrPoolThresholdStatus</code>	<code>jnxNatNotifications 1</code>	<p>The NAT address pool utilization threshold status trap indicates that the address pool utilization has either exceeded a certain set percentage or is lower than a set percentage.</p> <p><code>jnxNatTrapPoolName</code> is the name of the resource pool.</p> <p><code>jnxNatAddrPoolUtil</code> is the percentage of utilization of the address pool.</p>

**Related
Documentation**

- [NAT Pool Hit Table on page 641](#)
- [NAT Rule Hit Table on page 641](#)
- [Network Address Translation Resources—Monitoring MIB on page 639](#)
- [Source NAT Table on page 640](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Service Traps

The enterprise-specific Services PIC MIB provides traps for monitoring Adaptive Services (AS) PICs. [Table 337 on page 793](#) lists the supported traps.

Table 401: Supported Traps for Services PIC MIB

Object	Object Identifier	Description
<code>jnxSpSvcSetZoneEntered</code>	<code>jnxSPNotificationPrefix 1</code>	Indicates that an AS PIC has entered a more severe memory usage zone from a less severe memory usage zone. The zone entered is identified by <code>JnxSpSvcSetIfMemoryZone</code> .
<code>jnxSpSvcSetZoneExited</code>	<code>jnxSPNotificationPrefix 2</code>	Indicates that an AS PIC has exited a more severe memory usage zone to a less severe memory usage zone. The zone entered is identified by <code>JnxSpSvcSetIfMemoryZone</code> .
<code>jnxSpSvcSetCpuExceeded</code>	<code>jnxSPNotificationPrefix 3</code>	Indicates that an AS PIC has over 85% CPU usage. This trap is not supported on J Series Services Routers.
<code>jnxSpSvcSetCpuOk</code>	<code>jnxSPNotificationPrefix 4</code>	Indicates that an AS PIC has returned to less than 85% CPU usage. This trap is not supported on J Series Services Routers.

Related Documentation

- [Services PIC MIB on page 789](#)
- [jnxSpSvcSetTable on page 790](#)
- [jnxSpSvcSetSvcTypeTable on page 791](#)
- [jnxSpSvcSetIfTable on page 792](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific SNMP Version 1 Traps

The Junos OS supports enterprise-specific SNMP version 1 traps shown in [Table 402 on page 895](#). 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 Administration Library for Routing Devices*. To view the Juniper Networks enterprise-specific SNMP version 2 traps, see [“Juniper Networks Enterprise-Specific SNMP Version 2 Traps” on page 902](#). For more information about chassis traps, see [“Chassis Traps” on page 346](#).

Table 402: 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)

Table 402: 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
<i>Chassis MIB</i> (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.
	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.

Table 402: 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
	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.
	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							
<i>Configuration Management MIB (jnx-configmgmt.mib)</i>	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.
Link Notifications							

Table 402: 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
<i>Flow Collection Services MIB (jnx-coll.mib)</i>	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.

Table 402: 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
<i>Passive Monitoring MIB</i> (<i>jnx-pmonmib</i>)	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.
<i>SONET APS MIB</i> (<i>jnx-sonetaps.mib</i>)	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

Table 402: 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
<i>PING MIB</i> (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.
	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							
<i>BFD Experimental MIB</i> (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.

Table 402: 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
<i>LDP MIB</i> (jnx-ldp.mib)	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4	6	1	—	—	M, MX, T, and PTX Series routers.
	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4	6	2	—	—	M, MX, T, and PTX Series routers.
	jnxLdpSesUp	1.3.6.1.4.1.2636.4.4	6	3	—	—	M, MX, T, and PTX Series routers.
	jnxLdpSesDown	1.3.6.1.4.1.2636.4.4	6	4	—	—	M, MX, T, and PTX Series routers.
<i>MPLS MIB</i> (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	—	—	
<i>VPN MIB</i> (jnx-vpn.mib)	jnxVpnIfUp	1.3.6.1.4.1.2636.3.2.6	6	1	—	—	M, T, and MX Series routers.
	jnxVpnIfDown	1.3.6.1.4.1.2636.3.2.6	6	2	—	—	M, T, and MX Series routers.
	jnxVpnPwUp	1.3.6.1.4.1.2636.3.2.6	6	3	—	—	M, T, and MX Series routers.
	jnxVpnPwDown	1.3.6.1.4.1.2636.3.2.6	6	4	—	—	M, T, and MX Series routers.
RMON Alarms							
<i>RMON MIB</i> (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							

Table 402: 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
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 57](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific SNMP Version 2 Traps

The Junos OS supports the enterprise-specific SNMP version 2 traps shown in [Table 403 on page 902](#). 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 Administration Library for Routing Devices*. To view the Juniper Networks enterprise-specific SNMP version 1 traps, see [“Juniper Networks Enterprise-Specific SNMP Version 1 Traps” on page 894](#). For more information about chassis traps, see [“Chassis Traps” on page 346](#).

Table 403: 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

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>Chassis MIB</i> (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.
	jnxFruNotifAdminStatus		Notice		
	jnxFruNotifMismatch		Notice		
	jnxFruNotifOperStatus		Notice		
	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.
	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.

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
	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					
<i>Configuration Management MIB (jnx-cfgmgmt.mib)</i>	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					

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>Flow Collection Services MIB (jnx-coll.mib)</i>	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.
<i>PMON MIB (jnx-pmon.mib)</i>	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.

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>SONET APS MIB (jnx-sonetaps.mib)</i>	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					
<i>PING MIB (jnx-ping.mib)</i>	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					
<i>BFD Experimental MIB (jnx-bfd-exp.mib)</i>	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.

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>BGP4 V2 MIB</i> (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.
<i>DHCP MIB</i> (jnx-dhcp.mib)	jnxJdhcpLocalServer DuplicateClient	1.3.6.1.4.1.2636.3.6.1.6.1.13.1	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer InterfaceLimitExceeded	1.3.6.1.4.1.2636.3.6.1.6.1.13.2	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer InterfaceLimitAbated	1.3.6.1.4.1.2636.3.6.1.6.1.13.3	–	–	All devices running Junos OS.
	jnxJdhcpLocalServer Health	1.3.6.1.4.1.2636.3.6.1.6.1.13.4	–	–	All devices running Junos OS.
	jnxJdhcpRelayInterface LimitExceeded	1.3.6.1.4.1.2636.3.6.1.6.1.23.1	–	–	All devices running Junos OS.
	jnxJdhcpRelayInterface LimitAbated	1.3.6.1.4.1.2636.3.6.1.6.1.23.2	–	–	All devices running Junos OS.
<i>DHCPv6 MIB</i> (jnx-dhcpv6.mib)	jnxJdhcpv6LocalServer InterfaceLimitExceeded	1.3.6.1.4.1.2636.3.6.2.6.2.23.1	–	–	All devices running Junos OS.
	jnxJdhcpv6LocalServer InterfaceLimitAbated	1.3.6.1.4.1.2636.3.6.2.6.2.23.2	–	–	All devices running Junos OS.
	jnxJdhcpv6LocalServer Health	1.3.6.1.4.1.2636.3.6.2.6.2.23.3	–	–	All devices running Junos OS.
<i>LDP MIB</i> (jnx-ldp.mib)	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4.0.1	–	–	M, MX, T, and PTX Series routers.
	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4.0.2	–	–	M, MX, T, and PTX Series routers.
	jnxLdpSesUp	1.3.6.1.4.1.2636.4.4.0.3	–	–	M, MX, T, and PTX Series routers.
	jnxLdpSesDown	1.3.6.1.4.1.2636.4.4.0.4	–	–	M, MX, T and PTX Series routers.

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>MPLS MIB</i> (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, MX, T, and PTX 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, MX, T, and PTX 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, MX, T, and PTX 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, MX, T, and PTX Series routers.
	mplsLspInfoPathUp	1.3.6.1.4.1.2636.3.2.0.5	–	–	M, MX, T, and PTX Series routers.
<i>VPN MIB</i> (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.

Table 403: 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.
Access Authentication Methods MIB (jnx-js-auth.mib)	jnxJsFwAuthFailure	1.3.6.1.4.1.2636.3.39.1.2.1.0.1	–	–	J Series and SRX Series devices.
	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.
Network Address Translation Resources-Monitoring MIB (jnxNatMIB)	jnxJsNatAddrPoolThresholdStatus	1.3.6.1.4.1.2636.3.39.1.7.1.0.1	–	–	J Series and SRX Series devices.
	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

Table 403: Juniper Networks Enterprise-Specific Supported SNMP Version 2 Traps (*continued*)

Source MIB	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag	Supported On
<i>Network Address Translation MIB</i> (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.
<i>Security Screening Objects MIB</i> (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					
<i>RMON MIB</i> (jnx-rmon.mib)	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3.0.2	–	–	All devices running Junos OS.
SONET Alarms					
<i>SONET MIB</i> (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 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

Juniper Networks Enterprise-Specific Traps on EX Series Switches

EX Series switches support the following enterprise-specific traps:

- jnxVccpPortUp
- jnxVccpPortDown
- jnxBootFromBackup
- jnxFruInsertion

- Related Documentation**
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

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.
- **jnxl2aldMacMoveThreshold**—Generated when a MAC address move reaches the *threshold-count*.

For information about Juniper Networks enterprise-specific MIMSTP traps that are supported on MX Series routers, see “[Juniper Networks Enterprise-Specific MIMSTP Traps](#)” on page 620.

- Related Documentation**
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

VPN Traps

The enterprise-specific VPN MIB provides traps for monitoring VPNs. [Table 387 on page 874](#) lists supported VPN traps, VPNs, and circuit connection services.

Table 404: Supported VPN Traps, VPNs, and Circuit Connection Services

Object	Object Identifier	Layer 3 VPN	Layer 2 VPN	VPLS VPN	L2 Circuit	Circuit Cross-Connect	Optical VPN	Description
jnxVpnIfUp	jnxVpnMIBnotificatio 1	Yes	Yes	–	Yes	No	–	Indicates that the interface with the index jnxVpnIfIndex belonging to the jnxVpnIfVpnName of type jnxVpnIfVpnType went up.
jnxVpnIfDown	jnxVpnMIBnotificatio 2	Yes	Yes	–	Yes	No	–	Indicates that the interface with index jnxVpnIfIndex belonging to jnxVpnIfVpnName of type jnxVpnIfVpnType went down.
jnxVpnPwUp	jnxVpnMIBnotificatio 3	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with the index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went up.
jnxVpnPwDown	jnxVpnMIBnotificatio 4	No	Yes	Yes	Yes	No	–	Indicates that the pseudowire with index jnxVpnPwIndex belonging to jnxVpnPwVpnName of type jnxVpnPwVpnType went down.

Related Documentation

- [VPN MIB on page 863](#)
- [jnxVpnInfo on page 864](#)
- [jnxVpnTable on page 864](#)
- [jnxVpnIfTable on page 865](#)
- [jnxVpnPwTable on page 868](#)
- [jnxVpnRTTable on page 873](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

CHAPTER 85

Standard Traps

- [Standard SNMP Version 1 Traps on page 913](#)
- [Standard SNMP Version 2 Traps on page 916](#)
- [Standard SNMP Traps on EX Series Ethernet Switches on page 922](#)
- [Unsupported Standard SNMP Traps on page 922](#)

Standard SNMP Version 1 Traps

[Table 405 on page 913](#) 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 Administration Library for Routing Devices*.

Table 405: 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, <i>Conventions for Defining Traps for Use with the SNMP</i>	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							

Table 405: 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
RFC 1215, <i>Conventions for Defining Traps for Use with the SNMP</i>	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, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i>	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.
RMON Alarms							
RFC 2819a, <i>RMON MIB</i>	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							
<i>BGP 4 MIB</i>	bgpEstablished	1.3.6.1.2.1.15.7	6	1	—	—	M, T, MX, J, EX, PTX, and SRX for branch devices.
	bgpBackwardTransition	1.3.6.1.2.1.15.7	6	2	—	—	M, T, MX, J, EX, PTX, and SRX for branch devices.

Table 405: 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
<i>OSPF TRAP MIB</i>	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2	6	1	–	–	M, T, MX, J, EX, PTX, and SRX for branch devices.
	ospfNbrStateChange	1.3.6.1.2.1.14.16.2	6	2	–	–	M, T, MX, J, EX, PTX, and SRX for branch devices.
	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2	6	3	–	–	M, T, MX, J, EX, PTX, 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.

Table 405: 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
	ospflfStateChange	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, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i>	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.
RFC 6527, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)</i>	vrrpv3NewMaster	1.3.6.1.2.1.207	6	1	Warning	VRRPD_NEW_MASTER	M, MX and PTX
	vrrpv3ProtoError	1.3.6.1.2.1.207	6	2	Warning	VRRPD_V3_PROTO_ERROR	M, MX and PTX

- Related Documentation**
- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
 - [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
 - [Juniper Networks Enterprise-Specific MIBs on page 25](#)
 - [Standard SNMP MIBs Supported by Junos OS on page 9](#)
 - [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
 - [Managing Traps and Informs on page 58](#)

Standard SNMP Version 2 Traps

Table 406 on page 917 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 Administration Library for Routing Devices*.

Table 406: Standard Supported SNMP Version 2 Traps

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
Startup Notifications					
RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i>	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, <i>The Interfaces Group MIB</i>	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, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i>	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, <i>RMON MIB</i>	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.

Table 406: Standard Supported SNMP Version 2 Traps (*continued*)

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
Routing Notifications					
<i>BGP 4 MIB</i>	bgpEstablished	1.3.6.1.2.1.15.7.1	–	–	All devices running Junos OS.
	bgpBackwardTransition	1.3.6.1.2.1.15.7.2	–	–	All devices running Junos OS.
<i>OSPF Trap MIB</i>	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2.1	–	–	All devices running Junos OS.
	ospfNbrStateChange	1.3.6.1.2.1.14.16.2.2	–	–	All devices running Junos OS.
	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2.3	–	–	All devices running Junos OS.
	ospfIfConfigError	1.3.6.1.2.1.14.16.2.4	–	–	All devices running Junos OS.
	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.5	–	–	All devices running Junos OS.
	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2.6	–	–	All devices running Junos OS.
	ospfVirtIfAuthFailure	1.3.6.1.2.1.14.16.2.7	–	–	All devices running Junos OS.
	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2.8	–	–	All devices running Junos OS.
	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2.9	–	–	All devices running Junos OS.
	ospfTxRetransmit	1.3.6.1.2.1.14.16.2.10	–	–	All devices running Junos OS.
	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2.11	–	–	All devices running Junos OS.
	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2.13	–	–	All devices running Junos OS.
	ospfIfStateChange	1.3.6.1.2.1.14.16.2.16	–	–	All devices running Junos OS.

Table 406: Standard Supported SNMP Version 2 Traps (*continued*)

Defined in	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag	Supported On
MPLS Notifications					
RFC 3812, <i>Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base</i>	mplsTunnelUp				
	mplsTunnelDown				
	mplsTunnelRerouted				
	mplsTunnelReoptimized				
L3VPN Notifications					
RFC 4382, <i>MPLS/BGP Layer 3 Virtual Private Network (VPN)</i>	mplsL3VpnVrfUp				
	mplsL3VpnVrfDown				
	mplsL3VpnMulticast				
	mplsL3VpnMulticast				
	mplsL3VpnMulticast				
VRRP Notifications					
RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i>	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.
RFC 6527, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)</i>	vrrpv3NewMaster	1.3.6.1.2.1.207.0.1	Warning	VRRPD_NEW_MASTER	M and MX
	vrrpv3ProtoError	1.3.6.1.2.1.207.0.2	Warning	VRRPD_V3_PROTO_ERROR	M and MX

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

- [SNMP Version 2 MPLS Traps on page 920](#)
- [SNMP Version 2 L3VPN Traps on page 920](#)

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 Library for Routing Devices*.

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.

- **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.

SNMP Version 2 L3VPN Traps

The Junos OS also supports the following L3VPN SNMP version 2 traps defined in RFC 4382, *MPLS/BGP Layer 3 Virtual Private Network (VPN)*:

- **mplsL3VpnVrfUp**—Generated when:
 - No interface is associated with this VRF, and the first (and only first) interface associated with it has the **ifOperStatus** object transition to the **up** state.
 - Only one interface is associated with this VRF, and the **ifOperStatus** object of this interface transitions to the **up** state.

- Multiple interfaces are associated with this VRF, and the **ifOperStatus** object of all interfaces transitions to the **down** state, and the **ifOperStatus** object of the first of these interfaces transitions to the **up** state.
- **mplsL3VpnVrfDown**—Generated when:
 - One interface is associated with this VRF, and the **ifOperStatus** object of this interface transitions from the **up** state to the **down** state.
 - Multiple interfaces are associated with this VRF, and the **ifOperStatus** object of only one of these interfaces is in the **up** state, which then transitions to the **down** state.
 - The last interface with the **ifOperStatus** object in the **up** state is disassociated from a VRF.
- **mplsL3VpnVrfRouteMidThreshExceeded**—Generated when the number of routes contained by the specified VRF exceeds the value indicated by **mplsL3VpnVrfMidRouteThreshold**.

You can configure the **mplsL3VpnVrfMidRouteThreshold** value as follows:

```
[edit routing-instances <instance-name>]
user@R1# set routing-options maximum-paths <limit> threshold <threshold-value>
```

This configuration sets the **mplsL3VpnVrfMidRouteThreshold** value to **<threshold-value>** % of **<limit>**. This value can also be calculated as $(\text{<limit>} * \text{<threshold-value>}) / 100$.

- **mplsL3VpnVrfNumVrfRouteMaxThreshExceeded**—Generated when the number of routes contained by the specified VRF exceeds or attempts to exceed the maximum allowed value as indicated by **mplsL3VpnVrfMaxRouteThreshold**.

You can configure the **mplsL3VpnVrfMaxRouteThreshold** value as follows:

```
[edit routing-instances <instance-name>]
user@R1# set routing-options maximum-paths <limit> threshold <threshold-value>
```

To limit the frequency at which the **mplsL3VpnVrfNumVrfRouteMaxThreshExceeded** trap is generated, you can configure the **log-interval** statement as follows:

```
[edit routing-instances <instance-name>]
user@R1# set routing-options maximum-paths <limit> threshold <threshold-value>
log-interval <interval>
```

- **mplsL3VpnNumVrfRouteMaxThreshCleared**—Generated only after the number of routes contained by the specified VRF exceeds or attempts to exceed the maximum allowed value as indicated by **mplsVrfMaxRouteThreshold**, and then falls below this value.

Related Documentation

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

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*).

Related Documentation

- [Juniper Networks Enterprise-Specific SNMP Traps on page 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)
- [Configuring SNMP Trap Options and Groups on a Device Running Junos OS](#)
- [Managing Traps and Informs on page 58](#)

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 407 on page 923](#).

Table 407: 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	

Table 407: 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.
l3vpn mib.mib	mplsNumVrfSecIlglLblThrshExcd	Generated when the number of illegal label violations on a VRF table as indicated by mplsVpnVrfSecIllegalLblVltns has exceeded mplsVpnVrfSecIllegalLblRcvThrsh .
msdp mib.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.
ospf2 trap.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 .
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).

Table 407: Unsupported Standard SNMP Traps (*continued*)

MIB	Trap Name	Description
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 57](#)
- [Standard SNMP Traps Supported on Devices Running Junos OS on page 57](#)
- [Juniper Networks Enterprise-Specific MIBs on page 25](#)
- [Standard SNMP MIBs Supported by Junos OS on page 9](#)

PART 4

Index

- [Index on page 929](#)

Index

Symbols

#, comments in configuration statements.....	xxxviii
(), in syntax descriptions.....	xxxviii
< >, in syntax descriptions.....	xxxviii
[], in configuration statements.....	xxxviii
{ }, in configuration statements.....	xxxviii
(pipe), in syntax descriptions.....	xxxviii

A

AAA Objects MIB.....	25, 67
Textual Conventions.....	68
Access Authentication Objects MIB.....	25, 73
access authentication traps.....	70
Adaptive Services (AS) PIC.....	789
agent, SNMP.....	6
Alarm MIB.....	25, 77
Analyzer MIB.....	26, 81
Antivirus Objects MIB.....	26, 85
Antivirus Traps Objects Table.....	87, 877
apsChanConfigTable.....	808
apsChanStatusTable.....	809
apsConfigTable.....	804
apsStatusTable.....	805
ATM CoS MIB.....	26, 89
ATM MIB.....	26, 95
Auth Entry See Auth Entry MIB	
LSYS	
MIB.....	580

B

BFD MIB.....	26, 107
notification variables.....	108, 878
BGP4 V2 MIB.....	26, 105
braces, in configuration statements.....	xxxviii
brackets	
angle, in syntax descriptions.....	xxxviii
square, in configuration statements.....	xxxviii

C

Chassis Cluster MIB.....	27
Chassis Definitions for Router Model MIB.....	26, 352

Chassis Forwarding MIB.....	26, 113
Chassis MIB.....	27
jnxBoxAnatomy.....	118
jnxBoxKernelMemoryUsedPercent.....	346
jnxBoxSystemDomainType	346
jnxMIBs.....	118
jnxTraps.....	346, 879
overview.....	115
Class-of-Service MIB.....	27, 371
comments, in configuration statements.....	xxxviii
Configuration Management MIB.....	27, 375
conventions	
text and syntax.....	xxxvii
CoS	
MIB.....	27, 371
CPU See CPU MIB	
LSYS	
MIB.....	601
curly braces, in configuration statements.....	xxxviii
customer support.....	xxxix
contacting JTAC.....	xxxix

D

Destination Class Usage MIB.....	27, 381
DHCP MIB.....	27, 385
DHCPv6 MIB.....	27, 397
Digital Optical Monitoring MIB.....	28, 405
DNS Objects MIB.....	28, 411
documentation	
comments on.....	xxxix
Dynamic Flow Capture MIB.....	28, 354, 413
notification definitions.....	419, 885
notification variables.....	418, 424, 886

E

enterprise-specific traps, SNMP	
version 1.....	894
version 2.....	902
Ethernet MAC MIB.....	28, 421
Event MIB.....	28, 423
EX MAC Notification MIB.....	427
EX Series Ethernet Switches	
enterprise-specific traps.....	910
standard traps.....	922
Experimental MIB.....	28, 425, 823

F

Firewall MIB.....	28, 435
Flow Collection Services MIB.....	28, 439, 647

Flow Gate See Flow Gate MIB	
LSYS	
MIB.....	577
Flow Session See Flow Session MIB	
Flow session	
LSYS	
MIB.....	578
font conventions.....	xxxvii

G

Get requests, SNMP.....	3
-------------------------	---

H

Host Resources MIB.....	28, 445
-------------------------	---------

I

IDP attack log notification.....	797
IDP attack Table.....	796
IDP MIB.....	29, 795
ifChassisTable.....	450
Interface MIB.....	29, 447
IP Forward MIB.....	29, 453
IPsec Generic Flow Monitoring Object MIB.....	29, 457
Textual Conventions.....	458
IPsec Monitoring MIB.....	29, 473
IPsec Phase 1 IKE Tunnel Table.....	462
IPsec Phase 2 IKE Tunnel Table.....	465
IPsec Phase 2 Security Association Table.....	469
IPsec VPN Objects MIB.....	29, 481
Textual Conventions.....	482
IPv4 MIB.....	30, 485
IPv6 and ICMPv6 MIB.....	30

J

jnxBfdSessTable.....	107
jnxBgpM2PrefixCountersTable.....	105
jnxBoxAnatomy	118
jnxBoxKernelMemoryUsedPercent.....	346
jnxBoxSystemDomainType	346
jnxCollFileTable.....	442
jnxCollGlobalStats.....	439
jnxCollPicIfTable.....	440
jnxContainersTable	
M160 router.....	123
M20 router.....	122
M40 router.....	122
M40e router.....	132
M5 router.....	121
MX2010 router.....	130

MX2020 router.....	130
T320 router.....	132
T4000 router.....	133
T640 router.....	131
jnxContentsTable	
M20 router.....	135
MX2010 router.....	138
MX2020 router.....	146
T320 router.....	166
T640 router.....	162, 169
jnxCosInvQstatTable.....	371
jnxDCUsTable.....	381
jnxDcuStatsTable.....	382
jnxDfcCDTable.....	417
jnxDfcCSTable.....	414
jnxEventAvTable.....	423
jnxExperiment root branch.....	425, 823
jnxExVlanTable.....	842
jnxFilledTable.....	184
M20 router.....	185
MX2010 router.....	187
MX2020 router.....	193
T320 router.....	207
T4000 router.....	211
T640 router.....	203
jnxFruTable.....	258
M10 router.....	261
M160 router.....	268
M20 router.....	264
M40 router.....	274
M40e router.....	280
MX2010 router.....	285
MX2020 router.....	299
T4000 router.....	334
T640 router.....	324
jnxFwddProcess.....	113
jnxIfTable.....	447
jnxIkeTunnelTable.....	473
jnxIPSecSaTable.....	478
jnxIPSecTunnelTable.....	476
jnxJsAuthNotifications.....	74
jnxJsAuthTrapVars.....	74
jnxJsClusterObjectsTable.....	817
jnxJsDnsProxyDataObjects.....	411
jnxJsFwAuthStats.....	73
jnxJsIfMonTable.....	769
jnxJsIpSecTunnelTable.....	482
jnxJsLoadedCaCertTable.....	859
jnxJsLoadedLocalCertTable.....	860

jnxJsNatIfSrcPoolPortTable.....	636	jnxUserAAAAccessPool.....	71
jnxJsPolicyStatsTable.....	681	jnxUserAAAServerName.....	70
jnxJsScreenMonTable.....	774	jnxUserAAAStatTable.....	69
jnxJsScreenNotifications.....	786	jnxUtilCounter32Table.....	838
jnxJsScreenTrapVars.....	786	jnxUtilCounter64Table.....	838
jnxJsSPUMonitoringTable.....	816	jnxUtilIntegerTable.....	838
jnxLEDTTable.....	174	jnxUtilStringTable.....	839
M20 router.....	176	jnxUtilUintTable.....	839
MX2010 router.....	177	jnxVpnIfTable.....	865
MX2020 router.....	179	jnxVpnInfo.....	864
T320 router.....	183	jnxVpnPwTable.....	868
T640 router.....	182	jnxVpnRTTable.....	873
jnxMacStatsTable.....	421	jnxVpnTable.....	864
jnxMibs root branch.....	821		
jnxOperatingTable.....	215	L	
M20 router.....	218	L2ALD MIB.....	30, 537
MX2010 router.....	219	L2CP MIB.....	30, 541
MX2020 router.....	227	L2TP MIB.....	30, 543
T320 router.....	244	Layer 2 Control Protocol	
T4000 router.....	246	MIB.....	30, 541
T640 router.....	241	LDP	
jnxPfeNotifyGltTable.....	647	MIB.....	30, 557
jnxPfeNotifyTypeTable.....	649	traps.....	557, 558, 887
jnxPingCtlTable.....	664	disabling.....	558, 560, 887, 889
jnxPingLastTestResultTable.....	673	License MIB.....	30, 563
jnxPingProbeHistoryTable.....	671	Logical Systems MIB.....	30, 567
jnxPingResultsTable.....	668	LSYS	
jnxPMonFlowTable.....	660	MIB.....	30, 567, 577, 578, 580, 582, 584, 586, 588, 593, 597, 599, 601
jnxProducts root branch.....	819		
jnxRedundancyTable.....	250	M	
M20 router.....	252, 254	M120 router	
MX2010 router.....	253	MIB objects.....	354
T320 router.....	256	Management Information Base See MIBs	
T4000 router.....	257	Management Information MIB	
T640 router.....	255	jnxMibs.....	821
jnxRmonAlarmGetFailure.....	763	jnxProducts.....	819
jnxRmonAlarmTable.....	761	jnxServices.....	820
jnxRmonGetOk.....	763	jnxTraps.....	822, 886
jnxRpfStatsTable.....	759	manuals	
jnxScuStatsTable.....	371, 813	comments on.....	xxxix
jnxServices root branch.....	820	master agent, SNMP.....	6
jnxSonetAlarmsTable.....	799	MIBs	
jnxSpSvcSetIfTable.....	792	AAA Objects.....	25, 67
jnxSpSvcSetSvcTypeTable.....	791	Access Authentication Objects.....	25, 73
jnxSpSvcSetTable.....	790	Alarm.....	25, 77
jnxSyslogAvTable.....	834	Analyzer.....	26, 81
jnxSyslogTable.....	831	Antivirus Objects.....	26, 85
jnxTraceRouteCtlTable.....	835	ATM.....	26, 95
jnxTraps root branch.....	822, 886	ATM CoS.....	26, 89

BFD.....	26, 107	jnxServices.....	820
BGP4 V2.....	26, 105	jnxTraps.....	822, 886
Chassis.....	27, 115, 118, 346, 879	MIMSTP.....	31, 605
Chassis Cluster.....	27	MPLS.....	31, 625
Chassis Definitions for Router Model.....	26, 352	MPLS LDP.....	31, 623
Chassis Forwarding.....	26, 113	Multicast.....	15, 16, 25
Class-of-Service.....	27, 371	MX Series	
Configuration Management.....	27, 375	Virtual Chassis.....	433
Destination Class Usage.....	27, 381	NAT Objects.....	31, 633
DHCP	27, 385	NAT Resources-Monitoring.....	31
DHCPv6	27, 397	Optical Transport Network (OTN) Interface	
Digital Optical Monitoring.....	28, 405	Management	31, 643
DNS Objects.....	28, 411	OSPF.....	12
Dynamic Flow Capture.....	28, 354, 413	Packet Forwarding Engine.....	32, 647
Ethernet MAC.....	28, 421	Packet Mirror.....	32, 651
Event.....	28, 423	PAE Extension.....	32, 655
EX Series		Passive Monitoring.....	32, 659
Analyzer.....	26, 81	Ping.....	32, 663
PAE Extension.....	32, 655	Policy Objects.....	32, 679
SMI.....	431	Power Supply Unit.....	32, 685
Structure of Management Information		PPP.....	11, 32, 689
.....	35, 431	PPPoE.....	33, 717
Virtual Chassis.....	35, 433	Pseudowire ATM.....	33
VLAN.....	36, 841	Pseudowire TDM.....	33
Experimental.....	28, 425, 823	Real-Time Performance Monitoring.....	33, 751
Firewall.....	28, 435	Reverse-Path-Forwarding.....	33, 759
Flow Collection Services.....	28, 439, 647	RMON Events and Alarms	33, 761
Host Resources.....	28, 445	RPM.....	33, 751
IDP.....	29, 795	RSVP	34, 765
IDP Monitoring.....	795	Security Interface Extension Objects.....	34, 769
Interface.....	29, 447	Security Screening Objects.....	34, 773
IP Forward.....	29, 453	Services PIC.....	34, 789
IPsec Generic Flow Monitoring Object		SNMP IDP.....	29
.....	29, 457	IDP Monitoring.....	795
IPsec Monitoring.....	29, 473	SONET APS.....	34, 803
IPsec VPN Objects.....	29, 481	SONET/SDH Interface Management.....	34, 799
IPv4.....	30, 485	Source Class Usage.....	34, 813
IPv6 and ICMPv6.....	30	SPU Monitoring.....	35, 815
L2ALD.....	30, 537	Structure of Management	
L2CP	30, 541	Information.....	35, 819
L2TP.....	30, 543	Junos OS for J Series and SRX Series	
Layer 2 Control Protocol.....	30, 541	devices, for.....	35
LDP.....	30, 557	Subscriber.....	35, 825
License.....	30, 563	System Log.....	35, 831
Logical Systems.....	30	Traceroute.....	35, 835
LSys.30,567,577,578,580,582,584,586,588,593,597,599,601		Utility.....	35, 837
Management Information MIB		Virtual Chassis.....	35, 433
jnxMibs.....	821	VLAN.....	36, 841
jnxProducts.....	819		

- VPLS36, 849
 - BGP MIB.....36
 - Generic MIB.....36
 - LDP MIB.....36
- VPLS BGP.....855
- VPLS generic.....850
- VPLS LDP.....854
- VPN.....36, 863
- VPN Certificate Objects.....36, 859
- MIMSTP
 - MIB.....31, 605
- MPLS
 - enterprise-specific traps.....630, 892
 - MIB.....31
 - standard traps.....920
- MPLS LDP MIB.....31, 623
- MPLS MIB.....625
- Multicast MIB.....15, 16, 25
- MX 5 3D Universal Edge Router
 - MIB objects.....360
- MX10 3D Universal Edge Router
 - MIB objects.....360
- MX2010 3D Universal Edge Router
 - MIB objects.....357
- MX2020 3D Universal Edge Router
 - MIB objects.....356
- MX240 3D Universal Edge Router
 - MIB objects.....358
- MX40 3D Universal Edge Router
 - MIB objects.....359
- MX480 3D Universal Edge Router
 - MIB objects.....358
- MX80 3D Universal Edge Router
 - MIB objects.....359
- MX960 3D Universal Edge Router
 - MIB objects.....358
- N**
- NAT
 - LSYS
 - MIB.....582, 584, 586, 588, 593, 599
 - NAT cone bind See NAT cone bind MIB
 - LSYS
 - MIB.....597
 - NAT Objects MIB.....31, 633
 - NAT Pool Hit Table.....641
 - NAT Resources-Monitoring MIB.....31
 - NAT Rule Hit Table.....641
 - NAT trap definitions.....638, 892
 - NAT Trap Definitions.....642, 893
 - Network Address Translation Objects MIB See NAT Objects MIB
 - number of IKE Tunnels currently active462
 - nxContainersTable
 - M10 router.....121
- O**
- Optical Transport Network (OTN) Interface
 - Management MIB.31, 643
- OSPF MIB.....12
- P**
- Packet Forwarding Engine MIB.....32, 647
- Packet Mirror MIB.....32, 651
 - packet mirroring trap variables.....651
- packet mirroring traps.....653
- PAE Extension MIB.....32, 655
- parentheses, in syntax descriptions.....xxxviii
- Passive Monitoring MIB.....32, 659
- Ping MIB.....32, 663
- Policy Objects MIB.....32, 679
- Power Supply Unit MIB.....32, 685
- PPP MIB.....11, 32, 689
 - Global Attributes.....715
 - Interface Summary Counts.....708
 - IP NCP.....696
 - IPv6 NCP.....713
 - jnxPPPIpv6Table.....714
 - IPv6 NCP Layer
 - jnxPPPIpv6ConfigTable.....715
 - LCP Layer.....691
 - jnxPPPLinkConfigTable.....694
 - jnxPPPLinkStatusTable.....691
 - Multi-Link Layer.....703
 - jnxPPPMIPppBundleTable.....703
 - jnxPPPMIPppLinkBindTable.....707
 - jnxPPPMIPppLinkConfigTable.....704
 - jnxPPPMIPppNetworkConfigTable.....707
 - NCP Layer
 - jnxPPPIpConfigTable.....697
 - jnxPPPIpTable.....696
 - OSI NCP Layer.....698
 - jnxPPPOsiConfigTable.....700
 - jnxPPPOsiTable.....699
 - Session Attributes
 - jnxPppSessionTable.....701
 - Textual Conventions.....690

PPPoE MIB.....	33, 717
Interfaces Layer.....	722
jnxPPPoESubIfNextIndex.....	723
jnxPPPoESubIfQueueStatsTable.....	724
jnxPPPoESubIfTable.....	723
ServiceName Tables.....	725
jnxPPPoEServiceNameAciAriTable.....	727
jnxPPPoEServiceNameTable.....	726
jnxPPPoEServiceNameTableTable.....	726
Summary Counts.....	724
Textual Conventions.....	718
Underlying Interfaces Layer.....	718
jnxPPPoElfLockoutTable.....	721
jnxPPPoElfStatsTable.....	720
jnxPPPoElfTable.....	718
Pseudowire ATM MIB.....	33, 729
Pseudowire TDM MIB.....	33, 741
PSU MIB.....	32, 685
PTX5000 Services Gateway	
MIB objects.....	361

R

Real-Time Performance Monitoring MIB.....	33, 751
redundant adaptive services interfaces (rsp).....	793
Reverse-Path-Forwarding MIB.....	33, 759
RMON Events and Alarms MIB.....	33, 761
RPM MIB.....	33, 751
RSVP MIB.....	34, 765

S

Scan Engine Objects Table.....	86
Scan Statistics Objects Table.....	86
SCU, Source Class Usage See Source Class Usage	
MIB	
Security Interface Extension Objects MIB.....	34, 769
Security Policy Table.....	680
Security Profile See Security Profile MIB	
MIB.....	567
Security Screening Objects MIB.....	34, 773
Services PIC MIB.....	34, 789
traps.....	793, 894
Set requests, SNMP.....	3
SMI MIB for EX Series.....	431
SNMP	
agent.....	3, 6
architecture.....	3
enterprise-specific traps See SNMP traps	
manager.....	3
master agent.....	6

standard traps See SNMP traps	
standards documents.....	9
subagent.....	6
SNMP IDP MIB.....	795
SNMP traps.....	4
enterprise-specific	
version 1.....	894
version 2.....	902
EX Series Ethernet Switches.....	910, 922
MAC limit.....	911
MX960 3D Universal Edge Router.....	911
standard	
version 1.....	913
version 2.....	916
system logging severity levels.....	5
unsupported.....	922
SNMPv2	
MPLS traps.....	920
SONET APS MIB.....	34, 803
SONET Automatic Protection Switching	
MIB.....	34, 803
SONET/SDH Interface Management MIB.....	34, 799
Source Class Usage MIB.....	34, 813
Source NAT Statistics Table.....	640
Source NAT Table.....	634
SPU Monitoring MIB.....	35, 815
SRX 3600 Services Gateway	
MIB objects.....	365
SRX 5600 Services Gateway	
MIB objects.....	366
SRX100 Services Gateway	
MIB objects.....	367
SRX1400 Services Gateway	
MIB objects.....	364
SRX210 Services Gateway	
MIB objects.....	367
SRX220 Services Gateway	
MIB objects.....	368
SRX240 Services Gateway	
MIB objects.....	368
SRX3400 Services Gateway	
MIB objects.....	365
SRX550 Services Gateway	
MIB objects.....	368
SRX5800 Services Gateway	
MIB objects.....	366
SRX650 Services Gateway	
MIB objects.....	369

standard traps, SNMP	
version 1.....	913
version 2.....	916
standards documents	
SNMP and MIBs.....	10
Structure of Management Information	
MIB.....	35, 819
for EX Series.....	35, 431
Junos OS for J Series and SRX Series devices,	
for.....	35
subagent, SNMP.....	6
Subscriber MIB.....	35, 825
Textual Conventions.....	826
support, technical See technical support	
syntax conventions.....	xxxvii
System Log MIB.....	35, 831
system logging severity levels, SNMP traps.....	5
 T	
technical support	
contacting JTAC.....	xxxix
Traceroute MIB.....	35, 835
traps.....	902
definition.....	4
LDP.....	557, 558, 887
MPLS, enterprise-specific.....	630, 892
Services PIC MIB.....	793, 894
SNMP version 1 traps	
enterprise-specific.....	894
standard.....	913
SNMP version 2 traps	
enterprise-specific.....	902
standard.....	916
unsupported.....	922
See also SNMP traps	
 U	
unsupported standard SNMP traps.....	922
Utility MIB.....	35, 837
 V	
Virtual Chassis MIB.....	35, 433
VLAN MIB.....	36, 841
VPLS MIBs.....	36
overview.....	849
VPLS BGP MIB.....	855
VPLS Generic MIB.....	850
VPLS LDP MIB.....	854
VPN Certificate Objects MIB.....	36, 859
VPN MIB.....	36, 863

