EOS Command API Guide

Software Release 4.30.4M

November 9, 2023

http://www.arista.com **email:support@arista.com**

Contents

1 Arista Command API Fundamentals **2**

1.1 Configuring the Command API Agent . . . 3
1.1.1 Configuring a Certificate . . . 3
1.1.2 Configuring multiple VRFs . . . 3
1.2 The Spec . . . 4
1.2.1 Exploring the Command API . . . 4
1.2.2 An annotated example . . . 4
1.2.3 Command Specification . . . 8
1.2.4 Error Codes . . . 8
1.3 Using the Command API . . . 8
1.3.1 Unsupported commands . . . 9
1.3.2 Unconverted commands . . . 9
1.3.3 Streamed commands . . . 9
1.3.4 Using JSON-RPC . . . 9
1.4 On-box Usage of Command API . . . 10
1.4.1 Using EapiClientLib (requires no configuration) . . . 10
1.4.2 Other Methods (requires configuration) . . . 10
1.5 Using Websockets with Command API . . . 10
1.6 Client-Side Certificates . . . 11
1.7 Session Based Authorization . . . 13
1.7.1 User Login . . . 13
1.7.2 Command Requests . . . 13
1.7.3 User Logout . . . 13
1.7.4 Session Validation . . . 13
1.8 Versioning . . . 14
1.8.1 Incompatible Model . . . 14
1.8.2 Understanding Revisions And Versions . . . 15
1.8.3 Accessing a Command's Current Revision . . . 15
1.8.4 More Documentation On Revisions . . . 16

2 Command Output **17**

2.1 clear security pki certificate rotation ROTATION_ID . . . 18
2.2 ping mpls bgp labeled-unicast (ip (IP_ADDR_PREFIX | (IP_PREFIX_PREFIX | (IP_PREFIX_PREFIX_ADDR IP_PREFIX_PREFIX_MASK))) | (ipv6 (IP6_ADDR_PREFIX6 | IP6_PREFIX_PREFIX6)) [...] . . . 19
2.2.1 MplsPingHdr . . . 19
2.3 ping mpls bgp vpn (vrf (VRF | default)) (ip PREFIX) | (ipv6 PREFIX6) [...] . . . 22
2.4 ping mpls generic ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...] 22
2.5 ping mpls ldp ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...] . . 22
2.6 ping mpls p2mp mldp ROOTADDR ((lsp-id OPQVAL) | (multicast ipv4 SOURCEADDR GROUPADDR)) [...] . . . 22
2.7 ping mpls pseudowire ((ldp PW_NAME) | (vpls ldp INSTANCE_NAME group GROUP_NAME NEIGHBOR_ADDRESS)) [...] . . . 22

2.8 ping mpls rsvp session ((id SESSION_ID) | (name SESSION_NAME)) [...] . . . 23
2.9 ping mpls rsvp tunnel TUNNEL_NAME [...] . . . 23
2.10 ping mpls segment-routing ((((ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (ipv6 IPV6_ADDRESS) | (isis ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (isis ipv6 IPV6_ADDRESS))) | (ospf ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)))) [...] . . . 23
2.11 reset management dmf controller sync . . . 23
2.12 security pki certificate generate signing-request rotation ssl profile PROFILE_NAME key generate ((rsa RSA_BIT_LENGTH) | (ecdsa EC_CURVE_NAME)) [...] . . . 23
2.13 security pki certificate rotation commit ROTATION_ID . . . 23
2.14 security pki certificate rotation commit ssl profile PROFILE_NAME [...] . . . 24
2.15 security pki certificate rotation import ROTATION_ID . . . 24
2.16 session peer-supervisor CMD . . . 24
2.17 show ((ip bgp neighbors) | (ipv6 bgp peers)) ((received-routes) | (received-routes ((regex REGEX) | ((community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) })) | (access-list AS_PATH_LIST_NAME) | ((IPV4_ADDR | IPV6_ADDR)) | ((IPV4_ADDR_MASK | IPV6_ADDR_MASK))))) [...] . . . 25
2.17.1 BgpRouteHeader . . . 25
2.18 show ((ip bgp neighbors IP_PEER) | (ipv6 bgp peers (IP6_PEER | IP6_LL_PEER))) advertised-routes queued-withdrawals [...] . . . 36
2.18.1 BgpQueuedWithdrawals . . . 36
2.19 show ((ip bgp neighbors PEER) | (ipv6 bgp peers ((IP6_PEER | IP6_LL_PEER)))) (((routes | advertised-routes)) | ((routes | advertised-routes) ((regex REGEX) | ((community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) })) | (access-list AS_PATH_LIST_NAME) | ((IPV4_ADDR | IPV6_ADDR)) | ((IPV4_ADDR_MASK | IPV6_ADDR_MASK))))) [...] . . . 37
2.20 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) IP1 IP2 interface [...] . . . 38
2.20.1 MrouteSmInterface . . . 38
2.21 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) [...] . . . 40
2.21.1 GroupSm . . . 40
2.22 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) count [...] . . . 42
2.23 show ((ip pim) | (pim ipv4)) sparse-mode rp [...] . . . 42
2.23.1 PimRpCandidateSet . . . 43
2.24 show ((ip pim sparse-mode rp-hash) | (pim ipv4 sparse-mode rp hash)) GROUP [...] . . . 43
2.25 show ((ip pim sparse-mode upstream joins) | (pim ipv4 sparse-mode join)) [...] . . . 44
2.25.1 JoinPruneNeighbor . . . 44
2.26 show ((mac address-table) | mac-address-table) [...] . . . 45
2.26.1 MulticastMacAddressTable . . . 45
2.26.2 UnicastMacAddressTable . . . 45
2.27 show ((mac address-table) | mac-address-table) aging-time . . . 45
2.28 show ((mac address-table) | mac-address-table) count [...] . . . 45
2.28.1 VlanCount . . . 46
2.29 show ((mac address-table) | mac-address-table) dynamic local limit [...] . . . 46
2.29.1 LimitedVlanEntry . . . 46
2.30 show ((mac address-table) | mac-address-table) mlag-peer [...] . . . 46
2.31 show ((mac address-table) | mac-address-table) mlag-peer diff local-only . . . 46
2.32 show ((mac address-table) | mac-address-table) mlag-peer diff peer-only . . . 46
2.33 show ((mac address-table) | mac-address-table) multicast [...] . . . 46
2.33.1 IgmpSnoopingTableEntry . . . 47
2.34 show ((mac address-table) | mac-address-table) multicast brief [...] . . . 47
2.35 show ((mac address-table) | mac-address-table) notification host-flap detection window . . . 47
2.36 show ((mac address-table) | mac-address-table) notification host-flap logging . . . 47
2.37 show (arp | (ip arp)) [...] . . . 47
2.37.1 IpV4Neighbor . . . 47
2.38 show (arp | (ip arp)) resolve summary total [...] . . . 48

2.39 show (arp | (ip arp)) summary [...] . . . 48
2.40 show (arp | (ip arp)) | (ipv6 neighbors) remote [...] . . . 48
2.40.1 ArpBindingVlan . . . 48
2.41 show (bgp rpki cache) [...] . . . 49
2.41.1 CacheModel . . . 49
2.42 show (bgp rpki cache counters) [...] . . . 51
2.42.1 CacheCounterModel . . . 51
2.43 show (bgp rpki cache counters) errors [...] . . . 52
2.43.1 CacheRxErrorCounterModel . . . 52
2.43.2 CacheTxErrorCounterModel . . . 52
2.44 show (cli | routing-context) vrf . . . 52
2.45 show (ip | ipv6) attached-host route export [...] . . . 52
2.45.1 AttachedHostVlan . . . 53
2.46 show (ip | ipv6) bgp summary lldp [...] . . . 54
2.46.1 BgpSummary . . . 54
2.47 show (ip | ipv6) hardware ale routes acknowledgement [...] . . . 55
2.47.1 AcknowledgementRouteEntry . . . 56
2.48 show (ip | ipv6) hardware ale routes drop [...] . . . 56
2.48.1 VrfDropRoute . . . 56
2.49 show (ip | ipv6) hardware ale routes inconsistent [...] . . . 56
2.49.1 VrfRouteEntry . . . 56
2.50 show (ip | ipv6) hardware fib fec [...] . . . 57
2.50.1 FibFecInfo . . . 57
2.51 show (ip | ipv6) hardware fib summary . . . 59
2.51.1 AdjResourceOptimization . . . 59
2.52 show (ip | mac) access-lists payload alias . . . 59
2.52.1 PayloadSpec . . . 59
2.53 show ADDR_FAMILY dhcp relay access-list [...] . . . 61
2.53.1 IpAclList . . . 61
2.53.2 Ipv6AclList . . . 62
2.54 show ADDR_FAMILY dhcp relay counters . . . 62
2.54.1 GlobalCounters . . . 63
2.54.2 InterfaceCounter . . . 63
2.55 show ADDR_FAMILY helper-address . . . 64
2.55.1 ConfiguredInterface . . . 64
2.55.2 DefaultServersShowModel . . . 65
2.55.3 VssUnsupportedServersModel . . . 65
2.56 show APPLICATION profile . . . 65
2.56.1 FpgaApplicationInstanceInfo . . . 65
2.57 show APPLICATION profile all . . . 65
2.57.1 FpgaApplicationProfile . . . 66
2.58 show GNSS status . . . 67
2.58.1 Gnss . . . 67
2.59 show SYNCE . . . 68
2.59.1 SynceStatusInfo . . . 68
2.60 show aaa . . . 69
2.60.1 AaaUser . . . 69
2.61 show aaa (methods | method-lists) accounting . . . 69
2.61.1 AaaMethodListsAcctMethods . . . 69
2.62 show aaa (methods | method-lists) all . . . 70
2.62.1 AaaMethodListsAuthen . . . 70
2.62.2 AaaMethodListsAuthz . . . 70
2.63 show aaa (methods | method-lists) authentication . . . 70
2.64 show aaa (methods | method-lists) authorization . . . 70
2.65 show aaa authentication lockout . . . 71

2.65.1 AaaLockedUser . . . 71
2.66 show aaa counters . . . 71
2.67 show aaa sessions . . . 71
2.67.1 Sessions . . . 71
2.68 show adaptive-virtual-topology application-profile [...] . . . 72
2.68.1 AvtAppProfileByVrfModel . . . 72
2.69 show adaptive-virtual-topology flow [...] . . . 73
2.69.1 AvtFlowInfo . . . 73
2.70 show adaptive-virtual-topology path [...] . . . 74
2.70.1 AvtPathsByVrfModel . . . 74
2.71 show adaptive-virtual-topology path counters [...] . . . 75
2.71.1 AvtCountersByVrfModel . . . 75
2.72 show address locking . . . 76
2.72.1 IpLockingIntfEnabled . . . 76
2.73 show address locking counters . . . 76
2.73.1 IntfCounters . . . 77
2.73.2 ServerCounters . . . 77
2.74 show address locking servers . . . 77
2.74.1 IpLockingProtocol . . . 77
2.75 show address locking table ipv4 [...] . . . 77
2.75.1 IntfLeases . . . 77
2.76 show agent AGENT activity . . . 79
2.77 show agent AGENT memory allocations slab content SLABNAME SLABSIZE partial [...] . . . 80
2.77.1 SlabAllocatorContentPerThreadModel . . . 80
2.78 show agent AGENT memory detail . . . 80
2.79 show agent AGENT task scheduler [...] . . . 81
2.79.1 TaskScheduler . . . 81
2.80 show agent cpu [...] . . . 83
2.80.1 AgentCpuEntry . . . 83
2.81 show agent event history [...] . . . 84
2.81.1 LogEntryModel . . . 84
2.82 show agent memory [...] . . . 84
2.82.1 AgentMemoryEntry . . . 84
2.83 show agent memory allocations (delta DELTA) [...] . . . 84
2.83.1 AllocTrackTypeDiffModel . . . 84
2.84 show agent memory allocations [...] . . . 85
2.84.1 AllocTrackTypeModel . . . 85
2.85 show agent memory allocations heap summary [...] . . . 85
2.85.1 Mallinfo . . . 85
2.86 show agent memory allocations slab [...] . . . 86
2.86.1 SlabAllocatorAgentModel . . . 86
2.87 show agent memory allocations slab sort ((blocks (in-use | free | percent-free)) | (memory (in-use | free | percent-free | overhead | total | mmap | madvise | dirty))) . . . 87
2.88 show agent names [...] . . . 87
2.89 show agent oom scores [...] . . . 87
2.89.1 AgentOomScoresEntry . . . 87
2.90 show agent ping [...] . . . 87
2.90.1 AgentPingEntry . . . 88
2.91 show agent qtrace event tracking [...] . . . 88
2.91.1 QtEventTrackingEntryModel . . . 88
2.92 show agent qtrace size [...] . . . 88
2.92.1 AgentQtSizeSpecs . . . 88
2.93 show agent runnability [...] . . . 88
2.93.1 AgentRunnability . . . 89
2.94 show agent uptime [...] . . . 89

2.94.1 AgentUptimeEntry . . . 89
2.95 show agent warm-restart [...] . . . 89
2.95.1 ShowAgentWarmRestartEntry . . . 89
2.96 show aliases [...] . . . 89
2.96.1 Alias . . . 90
2.96.2 RegexAlias . . . 90
2.97 show application traffic recognition [...] . . . 90
2.97.1 ApplicationModel . . . 90
2.98 show application traffic recognition counters [...] . . . 90
2.98.1 ApplicationCountersModel . . . 90
2.99 show arp agent (ipv4 | ipv6) cache capacity [...] . . . 91
2.99.1 CapacityKinds . . . 91
2.100 show arp agent (ipv4 | ipv6) input [...] . . . 91
2.100.1 ArpInputEntryList . . . 91
2.101 show arp agent (ipv4 | ipv6) output [...] . . . 92
2.102 show arp monitor [...] . . . 92
2.102.1 MonitoredNexthopEntry . . . 92
2.102.2 MonitoredNexthopVrfSummary . . . 92
2.103 show arp monitor summary . . . 92
2.104 show arp | (ipv6 neighbors) bridged [...] . . . 92
2.105 show banner login . . . 93
2.106 show banner motd . . . 93
2.107 show bfd ((ip access-list) | (ipv6 access-list)) [...] . . . 93
2.108 show bfd (neighbors | peers) [...] . . . 94
2.108.1 BfdNeighborVrf . . . 94
2.109 show bfd (neighbors | peers) debug [...] . . . 98
2.109.1 Issue . . . 98
2.110 show bfd (neighbors | peers) history [...] . . . 99
2.110.1 PeerSnapshot . . . 99
2.111 show bfd (neighbors | peers) summary [...] . . . 102
2.111.1 SessionCountByType . . . 103
2.112 show bfd hardware acceleration . . . 103
2.113 show bfd hardware utilization [...] . . . 103
2.113.1 BfdHwResource . . . 103
2.114 show bfd peers protocol vxlan mlag primary . . . 104
2.114.1 VxlanVtepBfdStatus . . . 104
2.115 show bfd rbfd-stats . . . 104
2.116 show bfd sbfd-stats reflectors . . . 105
2.117 show bgp ((vpn-ipv4 | vpn-ipv6)) summary . . . 105
2.118 show bgp (AFI SAFI) ((IP | IP_PREFIX)) [...] . . . 105
2.119 show bgp (AFI SAFI) [...] . . . 106
2.120 show bgp (AFI SAFI) summary [...] . . . 106
2.121 show bgp (AFI SAFI) summary lldp [...] . . . 106
2.122 show bgp AFI unicast field-set FSNAME [...] . . . 106
2.123 show bgp access-list [...] . . . 106
2.124 show bgp auto-aggregation ipv4 [...] . . . 106
2.124.1 AutoAggregation . . . 106
2.125 show bgp auto-aggregation ipv6 [...] . . . 107
2.126 show bgp convergence [...] . . . 108
2.126.1 ShowBgpConvergenceVrfModel . . . 108
2.127 show bgp debug policy export ((((vrf-route filter-rcf ipv4 SAFI) | (vpn-ipv4)) PREFIX) | (((vrf-route filter-rcf ipv6 SAFI) | (vpn-ipv6)) PREFIX6)) [...] . . . 110
2.127.1 PolicyEvalResult . . . 110
2.127.2 RcfCodeUnitText . . . 118
2.128 show bgp debug policy export evpn (PREFIX | PREFIX6) [...] . . . 118

2.129 show bgp debug policy import ((((vpn-route filter-rcf vpn-ipv4) | (ipv4 SAFI)) PREFIX) | (((vpn-route filter-rcf vpn-ipv6) | (ipv6 SAFI)) PREFIX6)) rd RD [...] . . . 118
2.130 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...] . . . 118
2.131 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) (vpn-ipv4 | vpn-ipv6) (PREFIX | PREFIX6) rd RD [...] . . . 118
2.132 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) evpn route-type ip-prefix (PREFIX | PREFIX6) rd RD [...] . . . 118
2.133 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) flow-spec (ipv4 | ipv6) rule FLOWSPEC_RULE [...] . . . 119
2.134 show bgp debug policy outbound neighbor (ADDR | ADDR6) (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...] . . . 119
2.135 show bgp debug policy outbound neighbor (ADDR | ADDR6) (vpn-ipv4 | vpn-ipv6) (PREFIX | PREFIX6) rd RD [...] . . . 119
2.136 show bgp debug policy outbound neighbor (ADDR | ADDR6) evpn route-type ip-prefix (PREFIX | PREFIX6) rd RD [...] . . . 119
2.137 show bgp debug policy outbound neighbor (ADDR | ADDR6) flow-spec (ipv4 | ipv6) rule FLOWSPEC_RULE [...] . . . 119
2.138 show bgp debug policy redistribute (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...] . . . 119
2.139 show bgp evpn (route-type ((auto-discovery) | (mac-ip) | imet | (ethernet-segment) | (ip-prefix (ipv4 | ipv6 | PREFIX_V4 | PREFIX_V6)) | (smet | join-sync | leave-sync | spmsi))) internal [...] . . . 120
2.139.1 EvpnRouteEntry . . . 120
2.140 show bgp evpn [...] . . . 124
2.141 show bgp evpn host-flap . . . 124
2.141.1 BgpEvpnHostFlapEntry . . . 124
2.142 show bgp evpn instance [...] . . . 125
2.142.1 BgpEvpnInstance . . . 125
2.143 show bgp evpn sanity [...] . . . 127
2.143.1 EvpnSanityCategory . . . 127
2.144 show bgp evpn summary [...] . . . 127
2.145 show bgp flow-spec (ipv4 | ipv6 | vpn-ipv4 | vpn-ipv6) summary [...] . . . 127
2.146 show bgp flow-spec ipv4 [...] . . . 129
2.146.1 BgpFlowspecRules . . . 129
2.147 show bgp flow-spec ipv6 [...] . . . 131
2.148 show bgp flow-spec vpn-ipv4 [...] . . . 131
2.149 show bgp flow-spec vpn-ipv6 [...] . . . 131
2.150 show bgp instance [...] . . . 132
2.150.1 ShowBgpInstanceModel . . . 132
2.151 show bgp labeled-unicast forwarding [...] . . . 136
2.151.1 BgpLuForwardingTunnelTableEntry . . . 136
2.152 show bgp labeled-unicast forwarding raw [...] . . . 140
2.153 show bgp labeled-unicast tunnel [...] . . . 140
2.153.1 BgpLuTunnelTableEntry . . . 141
2.154 show bgp labeled-unicast tunnel raw [...] . . . 141
2.155 show bgp link-state [...] . . . 142
2.155.1 BgpLsRoutes . . . 142
2.156 show bgp link-state summary . . . 144
2.157 show bgp monitoring [...] . . . 145
2.157.1 BgpAdjRibInExportStatus . . . 145
2.157.2 BmpGlobalConfig . . . 145
2.157.3 BmpStation . . . 145
2.158 show bgp monitoring active [...] . . . 147
2.158.1 BmpActive . . . 147
2.159 show bgp monitoring active summary . . . 147

2.159.1 BmpActiveSummary . . . 147
2.160 show bgp monitoring internal scheduler [...] . . . 148
2.161 show bgp monitoring passive [...] . . . 148
2.161.1 BmpPassive . . . 148
2.162 show bgp monitoring passive summary . . . 148
2.162.1 BmpPassiveSummary . . . 148
2.163 show bgp monitoring summary . . . 149
2.163.1 BmpStationSummary . . . 149
2.164 show bgp mvpn-ipv4 (vrf VRF) route-type intra-as-ipmsi [...] . . . 150
2.164.1 MvpnRoutes . . . 150
2.165 show bgp mvpn-ipv4 (vrf VRF) route-type shared-tree [...] . . . 152
2.166 show bgp mvpn-ipv4 (vrf VRF) route-type source-active [...] . . . 152
2.167 show bgp mvpn-ipv4 (vrf VRF) route-type source-tree [...] . . . 152
2.168 show bgp mvpn-ipv4 route-type intra-as-ipmsi [...] . . . 152
2.169 show bgp mvpn-ipv4 route-type shared-tree [...] . . . 152
2.170 show bgp mvpn-ipv4 route-type source-active [...] . . . 152
2.171 show bgp mvpn-ipv4 route-type source-tree [...] . . . 153
2.172 show bgp neighbors (ADDR | ADDR6) rt-membership filter . . . 153
2.172.1 ShowRtMembershipNeighborFilterVrfModel . . . 153
2.173 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) ((routes | advertised-routes | (received-routes))) ((IP | IP_PREFIX)) [...] . . . 153
2.174 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) ((routes | advertised-routes | (received-routes))) [...] . . . 153
2.175 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) evpn (advertised-routes | routes | received-routes) [...] . . . 153
2.176 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) flow-spec ipv4 ROUTE_TYPE [...] . . . 154
2.177 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) flow-spec ipv6 ROUTE_TYPE [...] . . . 154
2.178 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) link-state ((routes | advertised-routes | (received-routes))) [...] . . . 154
2.179 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) path-selection ROUTE_TYPE [...] . . . 155
2.179.1 BgpDpsVrfRoutes . . . 155
2.180 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) rt-membership ROUTE_TYPE . . . 157
2.180.1 RtMembershipRoutes . . . 157
2.181 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) sr-te POLICY_FILTER [...] . . . 160
2.182 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv4 ((routes | advertised-routes | (received-routes))) PREFIX (rd RD_VAL) internal [...] . . . 161
2.182.1 MplsVpnRoutes . . . 161
2.183 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv4 ((routes | advertised-routes | (received-routes))) [...] . . . 164
2.184 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv6 ((routes | advertised-routes | (received-routes))) PREFIX (rd RD_VAL) internal [...] . . . 164
2.185 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv6 ((routes | advertised-routes | (received-routes))) [...] . . . 164
2.186 show bgp neighbors PEER vpls (advertised-routes | received-routes | routes) [...] . . . 165
2.186.1 BgpVplsRoute . . . 165
2.187 show bgp neighbors [...] . . . 167
2.188 show bgp neighbors history [...] . . . 167
2.188.1 NeighborHistoryVrfModel . . . 167
2.189 show bgp neighbors history socket . . . 168
2.189.1 NeighborHistorySocketVrfModel . . . 168
2.190 show bgp optimal-route-reflection position . . . 169
2.190.1 BgpOrrPosition . . . 169

2.191 show bgp path-selection [...] . . . 169
2.192 show bgp path-selection summary . . . 169
2.193 show bgp peer-group [...] . . . 170
2.193.1 BgpPeerGroups . . . 170
2.194 show bgp rcf [...] . . . 171
2.194.1 ShowAgentRcfFunctionModel . . . 171
2.195 show bgp route-flap-damping (AFI unicast) [...] . . . 173
2.195.1 BgpRfdVrfModel . . . 173
2.196 show bgp route-flap-damping policy [...] . . . 173
2.196.1 BgpRfdPolicy . . . 174
2.197 show bgp rpki roa ipv4 | ipv6 [...] . . . 174
2.197.1 Roa . . . 174
2.198 show bgp rpki roa summary . . . 174
2.198.1 CacheRoaSummaryModel . . . 174
2.198.2 TableRoaSummaryModel . . . 175
2.199 show bgp rpki roa table summary . . . 175
2.199.1 RoaTableSummaryModel . . . 175
2.200 show bgp rt-membership [...] . . . 175
2.201 show bgp rt-membership floodlist [...] . . . 175
2.201.1 ShowRtMembershipFloodlistVrfModel . . . 175
2.202 show bgp rt-membership origin-as (AS_PLAIN_NUM | AS_DOT_NUM) [...] . . . 175
2.203 show bgp rt-membership route-target ROUTE_TARGET [...] . . . 176
2.204 show bgp rt-membership summary . . . 176
2.205 show bgp scheduler [...] . . . 176
2.206 show bgp segment-routing prefix-segments . . . 176
2.206.1 PrefixSegment . . . 176
2.207 show bgp session tracker [...] . . . 176
2.207.1 BgpSessionTrackerEntry . . . 176
2.208 show bgp sr-te [...] . . . 177
2.209 show bgp sr-te community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) } [...] . . . 177
2.210 show bgp sr-te community-list COMM_LIST_NAME [...] . . . 177
2.211 show bgp sr-te endpoint ENDPOINT_VALUE color COLOR_VALUE distinguisher DISTINGUISHER_VALUE [...] . . . 177
2.212 show bgp sr-te endpoint ENDPOINT_VALUE path-selection color-vrf COLOR_VRF (avt AVT) router ROUTER_ID [...] . . . 177
2.213 show bgp sr-te summary [...] . . . 177
2.214 show bgp statistics [...] . . . 178
2.214.1 BgpStatistics . . . 178
2.214.2 BgpStatisticsGlobal . . . 180
2.215 show bgp summary [...] . . . 181
2.215.1 BgpSummaryAll . . . 181
2.216 show bgp traffic-policy field-set mappings [...] . . . 182
2.216.1 ShowTrafficPolicyFieldSetCommunityMappingsModel . . . 183
2.217 show bgp update-group [...] . . . 184
2.217.1 BgpUpdateGroups . . . 184
2.218 show bgp vpls [...] . . . 187
2.219 show bgp vpls auto-discovery [...] . . . 187
2.219.1 BgpVplsAdInstance . . . 187
2.220 show bgp vpls summary . . . 188
2.221 show bgp vpn-ipv4 PREFIX (rd RD_VAL) internal . . . 188
2.222 show bgp vpn-ipv4 [...] . . . 188
2.223 show bgp vpn-ipv6 PREFIX (rd RD_VAL) internal . . . 188
2.224 show bgp vpn-ipv6 [...] . . . 188
2.225 show boot stages log . . . 188

2.225.1 BootStagesLogEntryModel . . . 188
2.226 show boot-config . . . 189
2.227 show boot-extensions . . . 189
2.228 show bridge ((mac address-table) | mac-address-table) aging timeout . . . 190
2.229 show cdp neighbors . . . 191
2.229.1 CdpInterface . . . 191
2.230 show cfm alarm indication [...] . . . 192
2.230.1 MdAisModel . . . 192
2.231 show cfm continuity-check end-point [...] . . . 193
2.231.1 MdCcModel . . . 193
2.232 show cfm continuity-check end-point remote-defect [...] . . . 194
2.232.1 MdRdiModel . . . 194
2.233 show cfm counters [...] . . . 195
2.233.1 CfmIntfCounter . . . 195
2.234 show cfm end-point [...] . . . 195
2.234.1 MdModel . . . 195
2.235 show cfm intermediate-point . . . 196
2.236 show cfm measurement delay proactive [...] . . . 197
2.236.1 MdDmModel . . . 197
2.237 show cfm measurement loss proactive [...] . . . 199
2.237.1 MdLmModel . . . 199
2.238 show cfm measurement loss synthetic proactive [...] . . . 201
2.238.1 MdSlmModel . . . 201
2.239 show class-map [...] . . . 203
2.239.1 ClassMapModel . . . 203
2.240 show class-map type control-plane [...] . . . 204
2.241 show class-map type pbr [...] . . . 205
2.241.1 ClassMapModel . . . 205
2.242 show class-map type tapagg [...] . . . 213
2.243 show cli commands [({ hidden | functions | (truncate LENGTH) | verbatim })] . . . 213
2.244 show cli commands all-modes [({ hidden | functions | (truncate LENGTH) | verbatim })] . . . 213
2.245 show cli pid . . . 213
2.246 show clock [...] . . . 214
2.246.1 ClockSourceModel . . . 214
2.246.2 LocalTimeModel . . . 214
2.247 show command-tag [...] . . . 214
2.247.1 CommandTagStatus . . . 214
2.248 show configuration checkpoints . . . 214
2.248.1 Checkpoint . . . 215
2.249 show configuration consistency nat [...] . . . 215
2.249.1 ConfigConsistencyTableModel . . . 215
2.250 show configuration lock . . . 215
2.250.1 ConfigureLockModelUser . . . 215
2.251 show configuration lock history . . . 216
2.251.1 ConfigureLockModelHistoryInfo . . . 216
2.252 show configuration sessions [...] . . . 216
2.252.1 Session . . . 216
2.253 show configuration sessions memory . . . 217
2.253.1 SessionMemory . . . 217
2.254 show container-manager backup . . . 217
2.254.1 ContainerMgrBackupFiles . . . 217
2.255 show container-manager containers [...] . . . 217
2.255.1 ContainerMgrContainer . . . 217
2.256 show container-manager images [...] . . . 218
2.256.1 ContainerMgrImage . . . 218

2.257 show container-manager info . . . 219
2.258 show container-manager log CONTAINER_NAME . . . 219
2.259 show container-manager registry [...] . . . 219
2.259.1 ContainerMgrRegistry . . . 219
2.260 show cpu counters queue . . . 219
2.260.1 MakoCpuCountersQueueCoppClassCounts . . . 220
2.261 show cpu counters queue [...] . . . 221
2.261.1 EgressCpuCounters . . . 221
2.261.2 IngressCpuCounters . . . 221
2.262 show cvx [...] . . . 223
2.262.1 ClusterStatus . . . 223
2.262.2 ClusterStatusDebug . . . 224
2.262.3 SslProfileStatus . . . 224
2.263 show cvx connections [...] . . . 224
2.263.1 CvxConnection . . . 224
2.264 show cvx connections client (SYSTEMID | HOSTNAME) . . . 225
2.265 show cvx connections client (SYSTEMID | HOSTNAME) brief . . . 225
2.266 show cvx debug dirs [...] . . . 225
2.266.1 DirStatus . . . 225
2.267 show cvx debug pingpong . . . 225
2.268 show cvx file-replication . . . 226
2.268.1 Status . . . 226
2.269 show cvx file-replication NAME . . . 226
2.270 show cvx mounts . . . 226
2.270.1 CvxMounts . . . 226
2.271 show cvx mounts (HOSTNAME | SYSTEMID) . . . 227
2.272 show cvx service . . . 227
2.272.1 CvxService . . . 227
2.273 show cvx service NAME . . . 228
2.274 show cvx service peer [...] . . . 228
2.274.1 CvxPeerServiceStatus . . . 228
2.275 show daemon DAEMON_NAME ((ip access-list) | (ipv6 access-list) | (access-list)) [...] . . . 228
2.276 show daemon [...] . . . 229
2.276.1 DaemonAgent . . . 229
2.277 show dcbx [...] . . . 230
2.277.1 InterfaceDescription . . . 230
2.277.2 EtsTlvModel . . . 231
2.278 show debugging counter { COUNTERS } . . . 231
2.279 show debugging counters [...] . . . 231
2.280 show debugging inotify (QUERY | all) . . . 231
2.280.1 InotifyPids . . . 231
2.280.2 InotifyProc . . . 232
2.281 show debugging inotify file pid PID . . . 232
2.282 show debugging shark clients all . . . 232
2.283 show debugging smash clients all . . . 232
2.283.1 TableInfo . . . 232
2.284 show deep-inspection summary . . . 233
2.285 show dhcp server ((ipv4 leases) | (ipv6 leases) | leases) [...] . . . 234
2.285.1 DhcpServerShowLeasesModel . . . 234
2.286 show dhcp server [...] . . . 235
2.286.1 DhcpServerShowModel . . . 235
2.287 show direct-table [...] . . . 239
2.288 show direct-table errors . . . 239
2.288.1 TableState . . . 239
2.289 show direct-table state . . . 239

2.290 show direct-table table TABLE [...] . . . 239
2.290.1 ActionParameters . . . 240
2.291 show direct-table table TABLE counters . . . 240
2.291.1 EntryCounters . . . 240
2.292 show direct-table table TABLE entries . . . 241
2.292.1 EntryDetails . . . 241
2.292.2 ParameterType . . . 241
2.292.3 TableFieldType . . . 242
2.293 show direct-table table TABLE errors . . . 242
2.294 show direct-table table TABLE state . . . 242
2.295 show directflow . . . 242
2.296 show directflow counters [...] . . . 243
2.296.1 Flows . . . 243
2.297 show directflow detail [...] . . . 248
2.297.1 Details . . . 248
2.298 show directflow field-set . . . 248
2.298.1 DirectFlowEndPointList . . . 248
2.299 show directflow flows [...] . . . 249
2.300 show dns cache counters . . . 249
2.301 show dns domain . . . 249
2.302 show dot1q-tunnel [...] . . . 249
2.303 show dot1x all . . . 250
2.303.1 Dot1xInterfaceInformation . . . 250
2.303.2 Dot1xInterfaceDetails . . . 251
2.304 show dot1x all brief . . . 252
2.305 show dot1x all details . . . 252
2.306 show dot1x all statistics . . . 254
2.306.1 Dot1xDroppedCounterStats . . . 254
2.306.2 Dot1xInterfaceStats . . . 254
2.307 show dot1x all summary . . . 254
2.308 show dot1x hosts [...] . . . 254
2.308.1 Dot1xHosts . . . 254
2.309 show dot1x hosts blocked . . . 255
2.309.1 Dot1xBlockedMacEntry . . . 255
2.310 show dot1x hosts mac MAC [...] . . . 256
2.311 show dot1x interface INTFS . . . 256
2.312 show dot1x interface INTFS details . . . 257
2.313 show dot1x interface INTFS statistics . . . 257
2.314 show dot1x radius . . . 257
2.315 show dot1x supplicant [...] . . . 257
2.315.1 Dot1xSupplicant . . . 257
2.316 show dot1x vlan assignment group [...] . . . 258
2.317 show dynamic prefix-list [...] . . . 258
2.317.1 DynamicPrefixLists . . . 258
2.318 show e-tree vlan . . . 259
2.318.1 EtreeRoleModel . . . 259
2.319 show environment cooling [...] . . . 260
2.319.1 FanCollection . . . 260
2.320 show environment power . . . 262
2.320.1 Details . . . 262
2.320.2 PowerSupply . . . 262
2.321 show environment power detail . . . 263
2.322 show environment power voltage . . . 263
2.322.1 FruSlot . . . 263
2.323 show environment temperature [...] . . . 265

2.323.1 FruSlot . . . 265
2.324 show environment temperature transceiver [...] . . . 266
2.325 show etherchannel [...] . . . 267
2.325.1 PortChannel . . . 267
2.326 show event-handler [...] . . . 269
2.326.1 EventHandler . . . 269
2.327 show event-handler trigger counters [...] . . . 271
2.327.1 EventHandlerTriggerCounters . . . 271
2.328 show event-monitor lacp [...] . . . 271
2.328.1 LacpEventMonReasonDetails . . . 271
2.329 show extensions [...] . . . 273
2.329.1 Extension . . . 273
2.330 show feature-toggle . . . 274
2.330.1 ArToggleEntry . . . 274
2.331 show feature-toggle override status . . . 274
2.332 show file digest HASHTYPE URL . . . 274
2.333 show file information URL . . . 274
2.334 show file systems . . . 275
2.334.1 FileSystem . . . 275
2.335 show flow telemetry ipfix counters . . . 276
2.335.1 IpfixExporterCounter . . . 276
2.336 show flow telemetry ipfix flow-table [...] . . . 276
2.336.1 IpfixExporterFlow . . . 276
2.337 show flow telemetry ipfix hostname . . . 277
2.337.1 IpfixHostname . . . 277
2.338 show flow telemetry ipfix option-data . . . 277
2.338.1 IpfixExporterOptionData . . . 278
2.339 show flow telemetry ipfix template . . . 278
2.339.1 IpfixExporterTemplate . . . 278
2.340 show flow telemetry postcard counters . . . 279
2.341 show flow telemetry postcard flow-table [...] . . . 279
2.342 show flow telemetry postcard hostname . . . 279
2.343 show flow telemetry postcard option-data . . . 279
2.344 show flow telemetry sflow counters . . . 279
2.345 show flow telemetry sflow flow-table [...] . . . 279
2.346 show flow telemetry sflow hostname . . . 279
2.347 show flow telemetry sflow option-data . . . 279
2.348 show flow tracking (sampled | hardware | (telemetry inband)) ipfix options-table [...] . . . 279
2.348.1 TrackerOptTableModel . . . 280
2.349 show flow tracking (sampled | hardware | (telemetry inband)) ipfix template [...] . . . 280
2.349.1 TrackerTemplateModel . . . 280
2.350 show flow tracking (sampled | hardware | mirror-on-drop | (telemetry inband)) [...] . . . 284
2.350.1 EncapTypeModel . . . 284
2.350.2 FtModel . . . 284
2.350.3 FtrTrafficTypeModel . . . 287
2.351 show flow tracking hardware counters [...] . . . 288
2.351.1 TrackerCounters . . . 288
2.352 show flow tracking hardware flow-table [...] . . . 290
2.352.1 TrackerModel . . . 290
2.353 show flow tracking hardware flow-table [...]_1 . . . 292
2.353.1 TrackerModel . . . 292
2.354 show flow tracking mirror-on-drop counters [...] . . . 294
2.354.1 TrackerModel . . . 294
2.355 show flow tracking mirror-on-drop counters drop-reason [...] . . . 296
2.356 show flow tracking mirror-on-drop flow-table [...] . . . 296

2.357 show flow tracking sampled counters [...] . . . 297
2.357.1 HardwareOffloadCounters . . . 297
2.357.2 SampleCounters . . . 297
2.357.3 TrackerCounters . . . 298
2.358 show flow tracking sampled counters debug . . . 299
2.358.1 PacketErrorCounter . . . 299
2.359 show flow tracking sampled counters interface errors [...] . . . 300
2.359.1 PerInterfaceFlowErrCounters . . . 300
2.360 show flow tracking sampled flow-table [...] . . . 301
2.360.1 TrackerModel . . . 301
2.361 show flow tracking sampled flow-table hardware [...] . . . 305
2.361.1 HoTrackerModel . . . 305
2.362 show flow tracking sampled flow-table hardware counters [...] . . . 306
2.362.1 HoTrackerCounterModel . . . 306
2.363 show flow tracking telemetry inband counters [...] . . . 306
2.364 show flow tracking telemetry inband flow-table [...] . . . 306
2.365 show flow-spec (ipv4 | ipv6) summary [...] . . . 307
2.365.1 VrfFlowspecRuleSummary . . . 307
2.366 show flow-spec ipv4 [...] . . . 308
2.366.1 VrfFlowspecRule . . . 308
2.367 show flow-spec ipv6 [...] . . . 311
2.368 show flowcontrol [...] . . . 311
2.368.1 FlowControlData . . . 311
2.369 show forwarding destination [...] . . . 311
2.369.1 EgressInterface . . . 311
2.369.2 PacketDestinationDetailModel . . . 311
2.370 show forwarding destination ingress-interface <ingressIntf> raw <rawPacket> [...] . . . 312
2.371 show fpga applications [...] . . . 312
2.372 show fpga applications compatibility . . . 312
2.372.1 FpgaApplicationsCompatibility . . . 312
2.373 show fpga applications profile [...] . . . 312
2.374 show fpga switch ((mac address-table) | mac-address-table) multicast [...] . . . 312
2.374.1 EthGroup . . . 312
2.375 show fpga switch [...] . . . 313
2.375.1 FpgaSwitchCore . . . 313
2.376 show fpga switch counters [...] . . . 313
2.376.1 MakoDebugCounterValue . . . 313
2.377 show fpga switch ip route [...] . . . 314
2.377.1 FpgaSwitchCoreRoutes . . . 314
2.378 show fpga switch mapping [...] . . . 314
2.378.1 PortMappings . . . 314
2.379 show fpga switch multicast ip route [...] . . . 315
2.379.1 FpgaSwitchMulticastCoreRoutes . . . 315
2.380 show fpga switch parity event [...] . . . 315
2.380.1 ParityEvent . . . 316
2.381 show fpga switch shared resources mbb [...] . . . 317
2.381.1 MbbTables . . . 317
2.382 show fpga switch shared resources operational status (nexthop | acl) [...] . . . 318
2.382.1 OperationalStatus . . . 318
2.383 show fpga switch shared resources operational status nat [...] . . . 322
2.384 show gre tunnel static [...] . . . 322
2.384.1 L3IntfTunnelTableEntry . . . 322
2.385 show hardware abstraction table NAMESPACE TABLE_NAME [...] . . . 323
2.385.1 TableConfigModel . . . 323
2.386 show hardware abstraction table NAMESPACE TABLE_NAME summary . . . 323

2.386.1 TableConfigSummaryModel . . . 323
2.387 show hardware asset-tag [...] . . . 324
2.387.1 Asset . . . 324
2.387.2 Missing . . . 324
2.388 show hardware capacity [...] . . . 324
2.388.1 TableEntry . . . 324
2.389 show hardware capacity alert threshold . . . 325
2.389.1 TableThreshold . . . 325
2.390 show hardware clock frequency sink [...] . . . 325
2.390.1 FreqSinkInfo . . . 325
2.391 show hardware clock frequency source [...] . . . 325
2.391.1 FreqSourceInfo . . . 325
2.392 show hardware clock frequency topology . . . 326
2.392.1 SinkInfo . . . 326
2.393 show hardware clock strobe sink [...] . . . 326
2.393.1 StrobeSinkInfo . . . 326
2.394 show hardware clock strobe source [...] . . . 326
2.394.1 StrobeSourceInfo . . . 326
2.395 show hardware clock strobe topology . . . 326
2.396 show hardware clocking . . . 326
2.396.1 ClockPortStatus . . . 327
2.397 show hardware counter drop [...] . . . 328
2.397.1 DropEvents . . . 328
2.398 show hardware counter events . . . 328
2.398.1 CounterEvent . . . 329
2.399 show hardware counter feature . . . 330
2.399.1 CounterFeatureInfo . . . 333
2.400 show hardware eeprom [...] . . . 334
2.400.1 Component . . . 334
2.401 show hardware external alarm-input . . . 335
2.401.1 ExternalAlarm . . . 335
2.402 show hardware external alarm-input ALARM_ID . . . 335
2.403 show hardware forwarding profile . . . 335
2.404 show hardware fpga detail . . . 335
2.404.1 FpgaHardwareDetails . . . 335
2.405 show hardware fpga error . . . 337
2.405.1 FpgaCrcErrorDetails . . . 337
2.406 show hardware logical-port (pool | POOL_RANGE) status . . . 338
2.406.1 PoolStatus . . . 338
2.407 show hardware pll [...] . . . 338
2.407.1 ClockModel . . . 338
2.408 show hardware resource [...] . . . 338
2.408.1 AgentName . . . 338
2.409 show hardware speed-group | SPEED-GROUP configurations . . . 339
2.409.1 SpeedGroupCap . . . 339
2.410 show hardware speed-group | SPEED-GROUP status . . . 339
2.410.1 SpeedGroupStatus . . . 339
2.411 show hardware system forwarding-chips . . . 340
2.411.1 ForwardingChip . . . 340

2.412 show hardware tcam profile ((PROFILE ((feature (((acl vlan ip) | (acl vlan ipv6) | (acl vlan ipv6 egress) | (acl subintf ip) | (acl subintf ipv6) | (acl subintf mac) | (acl port ip) | (acl port ipv6) | (acl port mac) | (acl gre tunnel interface ip) | (acl port ip egress) | (acl port ip egress disabled) | (acl port ip egress mpls-tunnelled-match) | (acl port ipv6 egress) | (acl port ipv6 source-only egress) | (acl vlan ipv6 source-only egress) | (acl port mac egress) | (acl vlan ip egress) | (acl port ip ingress) | (acl vlan ip ingress) | (acl port ipv6 ingress) | (acl vlan ipv6 ingress) | (acl port mac ingress) | (acl port ipv6 source-only ingress) | (acl tunnel interface ip) | (pbr ip) | (pbr ipv6) | (pbr mpls) | (pbr subintf ip) | (pbr subintf ipv6) | (pbr tunnel interface ip) | (tunnel ipv4) | (tunnel vxlan) | (tunnel vxlan multicast mlag) | (tunnel vxlan multi-homing) | (tunnel vxlan multi-homing domain remote) | (tunnel vxlan routing) | (tunnel vxlan routing v6-underlay) | (tunnel vxlan vni policing) | (qos ip) | (qos ipv6) | (qos subintf ip) | (qos subintf ipv6) | (qos port ip) | (qos port ipv6) | (qos vlan ip) | (qos vlan ipv6) | (qos port ip egress) | (qos vlan ip egress) | (qos vlan ipv6 egress) | (qos mac) | (tapagg port) | (tapagg ip) | (tapagg ipv6) | (tapagg mac) | (tapagg tunnel termination) | (mirror ip) | (mirror ipv6) | (mirror mac) | (mirror mpls-over-gre ip) | (mirror mpls-over-gre ipv6) | (mirror port egress rate-limit) | (hash mpls-over-gre inner ipv4) | (hash mpls-over-gre inner ipv6) | (evpn mpls irb) | (evpn mpls multi-homing) | flow | (flow mac) | (flow mac vlan-rewrite) | (flow http) | (flow mac-ipv4-l4) | (flow ipv6) | (flow ipv6 l4) | (flow ipv6 vlan-tag) | cbf | (flex-route) | (counter multicast ipv4) | mpls | (mpls pop ingress) | (counter lfib) | (counter vlan-interface ingress) | (counter vlan traffic-class) | (counter vtep decap) | (flow-spec port ipv4) | (flow-spec port ipv6) | (l2-protocol forwarding) | ipsec | (ipsec egress padding-removal) | (traffic-policy cpu ipv4) | (traffic-policy cpu ipv6) | (traffic-policy port ipv4) | (traffic-policy port ipv6) | (traffic-policy port ipv4 egress) | (traffic-policy port ipv6 egress) | (traffic-policy vlan-bridged ipv4) | (traffic-policy vlan-bridged ipv6) | (traffic-policy vlan-interface ipv4) | (traffic-policy vlan-interface ipv6) | (segment-security ipv4) | (segment-security ipv6) | (forwarding-destination) | (forwarding-destination mpls) | (tcp-mss-ceiling ip) | (macsec-proxy) | (interface-policing) | (telemetry postcard policy ipv4) | (telemetry postcard policy ipv6) | (flow tracking sampled ipv4) | (ptp e2etransparent one-step) | (vlan editing extended-64) | (vrf selection) | (vrf selection extended) | cfm | rfc2544 | (mlag egress filter disabled) | (unicast rpf lookup-vrf) | (segment-routing ipv6) | (sflow filtered-sampling) | (sflow subintf) | (vpls irb) | (private-vlan) | (storm-control subintf)))) | detail)) | detail) 342
2.412.1 ProfileModel . . . 343
2.413 show hardware tcam profile PROFILE . . . 343
2.414 show hardware-license tpm . . . 344
2.415 show health-monitor counters . . . 344
2.415.1 HmNhgCounterEntry . . . 344
2.416 show health-monitor probe delay [...] . . . 345
2.416.1 HmPbEntry . . . 345
2.417 show health-monitor status . . . 345
2.417.1 HealthMonitorProbeStatus . . . 345
2.418 show health-monitor status detail . . . 345
2.418.1 HealthMonitorServerProbeStatusDetail . . . 345
2.419 show health-monitor summary . . . 346
2.419.1 HealthMonitorLbSummary . . . 346
2.420 show health-monitor timestamps . . . 346
2.420.1 HmNhgTimestampEntry . . . 346
2.421 show hostname . . . 347
2.422 show hosts . . . 347
2.422.1 NameServerConfig . . . 347
2.422.2 IpAddresses . . . 347
2.423 show hsc arp sources SOURCE [...] . . . 347
2.423.1 HscArpSourceModel . . . 347
2.424 show hsc certificate . . . 347
2.425 show hsc logical router [...] . . . 348
2.425.1 HscLogicalRouterModel . . . 348

2.426 show hsc logical switch [...] . . . 348
2.426.1 HscLogicalSwitchModel . . . 348
2.427 show hsc physical locator [...] . . . 349
2.428 show hsc physical locator vni VNI [...] . . . 349
2.429 show hsc physical port [...] . . . 349
2.429.1 HscPhysicalPortModel . . . 349
2.430 show hsc physical switch [...] . . . 349
2.430.1 HscPhysicalSwitchModel . . . 349
2.431 show hsc status [...] . . . 351
2.431.1 HscAgentStatusModel . . . 351
2.431.2 OvsdbManagerModel . . . 351
2.431.3 SslProfileModel . . . 351
2.432 show hsc status ovsdb [...] . . . 352
2.433 show idprom (transceiver | interface) [...] . . . 352
2.433.1 InterfacesTransceiverIdpromPages . . . 352
2.434 show igmp snooping querier [...] . . . 352
2.434.1 IgmpSnoopingVlanQuerier . . . 353
2.435 show igmp snooping querier counters [...] . . . 354
2.435.1 IgmpSnoopingVlanQuerierCounters . . . 354
2.436 show igmp snooping querier membership [...] . . . 355
2.436.1 IgmpSnoopingVlanQuerierMembership . . . 355
2.437 show igmp snooping querier membership vlan VLAN_ID group IPADDR . . . 356
2.438 show igmp snooping querier status [...] . . . 356
2.439 show installed-extensions . . . 357
2.439.1 InstallationStatus . . . 357
2.440 show interfaces (capabilities | hardware) | (capabilities | hardware (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 358
2.440.1 InterfaceCapabilitiesData . . . 358
2.441 show interfaces (capabilities | hardware default) | (capabilities | hardware default (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 361
2.441.1 InterfaceCapabilitiesDefaultData . . . 361
2.442 show interfaces (counters bins) | (counters bins (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 362
2.442.1 InterfaceInOutBinsCounters . . . 362
2.443 show interfaces (counters errors) | (counters errors (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 362
2.443.1 ErrorCounters . . . 362
2.444 show interfaces (counters half-duplex) | (counters half-duplex (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 363
2.444.1 HalfDuplexErrorCounters . . . 363
2.445 show interfaces (counters ingress acl drop) | (counters ingress acl drop (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 363
2.445.1 InterfacesCounterAclDrop . . . 363
2.446 show interfaces (error-correction) | (error-correction (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 364
2.446.1 InterfaceErrorCorrectionStatus . . . 364

2.447 show interfaces (flow-control | flowcontrol) | (flow-control | flowcontrol (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 365
2.448 show interfaces (l2 mtu) | (l2 mtu (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 365
2.448.1 L2Mtu 365
2.449 show interfaces (phy detail) | (phy detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 366
2.449.1 InterfacePhyStatusesDetailed 366
2.450 show interfaces (phy diag error-correction histogram) | (phy diag error-correction histogram (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . 427
2.450.1 IntfFecHistogramPhy 427
2.451 show interfaces (phy diag test pattern) | (phy diag test pattern (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 429
2.451.1 IntfTestPatternStatus 429
2.452 show interfaces (phy diag test pattern counters) | (phy diag test pattern counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . 430
2.452.1 IntfTestPatternCounters 430
2.453 show interfaces (phy diag test pattern counters detail) | (phy diag test pattern counters detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 431
2.453.1 IntfTestPatternDetail 431
2.454 show interfaces (phy modulation) | (phy modulation (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 432
2.454.1 InterfacePhyLineRate 432
2.454.2 InterfacePhyModulation 432
2.455 show interfaces (phy precoding) | (phy precoding (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 433
2.455.1 InterfacePrecoding 433
2.456 show interfaces (phy skew) | (phy skew (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 433
2.456.1 InterfacesPhySkew 434
2.457 show interfaces (phy skew persistent) | (phy skew persistent (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 434
2.457.1 InterfacesPhySkewPstore 434
2.458 show interfaces (priority-flow-control) | (priority-flow-control (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] 435
2.458.1 PfcFrameCountersModel 435
2.458.2 PfcIntfStatusModel 435
2.458.3 PfcStatusModel 436
2.458.4 PfcWatchdogStatusModel 436
2.459 show interfaces (priority-flow-control counters) | (priority-flow-control counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . 437

2.460 show interfaces (priority-flow-control counters detail) | (priority-flow-control counters detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 437
2.460.1 PfcIntfDetailedCountersModel . . . 437
2.461 show interfaces (priority-flow-control counters watchdog) | (priority-flow-control counters watchdog (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 437
2.461.1 PfcWatchdogIntfCountersModel . . . 438
2.462 show interfaces (priority-flow-control counters watchdog drop) | (priority-flow-control counters watchdog drop (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 438
2.462.1 PfcWatchdogIntfEgressDropCountersModel . . . 438
2.463 show interfaces (priority-flow-control status) | (priority-flow-control status (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 439
2.464 show interfaces (status errdisabled) | (status errdisabled (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 439
2.464.1 InterfaceCauseModel . . . 439
2.465 show interfaces (tap | tool) | (tap | tool (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 441
2.465.1 TapProperty . . . 441
2.465.2 ToolProperty . . . 442
2.466 show interfaces Fabric priority-flow-control . . . 443
2.467 show interfaces [...] . . . 444
2.467.1 InterfaceStatus . . . 444
2.467.2 ApplicationInterfaceStatus . . . 446
2.467.3 EthInterfaceStatus . . . 448
2.467.4 EthPhyInterfaceStatus . . . 451
2.467.5 InternalRecircInterfaceStatus . . . 452
2.467.6 LagEthInterfaceStatus . . . 454
2.467.7 InterfaceConfigSanity . . . 455
2.467.8 TunnelIntfConfigSanity . . . 457
2.467.9 TunnelIntfStatus . . . 458
2.467.10 VlanInterfaceStatus . . . 460
2.467.11 VxlanInterfaceStatus . . . 462
2.468 show interfaces all hardware default . . . 466
2.469 show interfaces cable detail [...] . . . 467
2.469.1 BaseTCableTestStatus . . . 467
2.470 show interfaces config-sanity [...] . . . 468
2.471 show interfaces counters discards [...] . . . 468
2.471.1 InterfaceCounterDiscards . . . 468
2.472 show interfaces counters drop-precedence [...] . . . 468
2.472.1 IntfDropPrecedenceCounters . . . 468
2.473 show interfaces counters egress acl drop [...] . . . 468
2.473.1 EgressAclDropCounters . . . 469
2.474 show interfaces counters egress mtu [...] . . . 469
2.474.1 EgressMtuCounters . . . 469
2.475 show interfaces counters ip [...] . . . 469
2.475.1 InterfaceIpv4Ipv6Counters . . . 469
2.476 show interfaces counters ip rates [...] . . . 469

2.476.1 InterfaceIpv4Ipv6CounterRates . . . 470
2.477 show interfaces counters queue [...] . . . 471
2.477.1 EgressQueueCounters . . . 471
2.477.2 IngressQueueCounters . . . 472
2.477.3 IngressVoqCounters . . . 472
2.478 show interfaces counters queue drop-precedence [...] . . . 472
2.478.1 IntfDropPrecedenceCounters . . . 472
2.479 show interfaces counters queue drops [...] . . . 472
2.479.1 QueueTrafficClassDropsCounters . . . 473
2.480 show interfaces counters queue drops unicast [...] . . . 473
2.480.1 EgressQueueDropCounters . . . 474
2.481 show interfaces counters queue ingress [...] . . . 474
2.482 show interfaces counters queue rates [...] . . . 474
2.483 show interfaces counters rates [...] . . . 475
2.483.1 InterfaceCountersRateBase . . . 475
2.483.2 FabricIntfCountersRateModel . . . 475
2.483.3 FabricPairCountersRateModel . . . 475
2.483.4 InterfaceCountersRate . . . 476
2.483.5 VlanInterfaceCountersRate . . . 476
2.483.6 VxlanInterfaceCountersRate . . . 476
2.484 show interfaces counters traffic-class [...] . . . 476
2.484.1 InterfaceTrafficClassesCounters . . . 477
2.485 show interfaces counters vni [...] . . . 477
2.485.1 VirtualInterfaceCountersBase . . . 477
2.485.2 VxlanInterfaceCounters . . . 477
2.486 show interfaces counters vni rates [...] . . . 478
2.487 show interfaces counters | (counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 479
2.487.1 InterfaceCountersBase . . . 479
2.487.2 InterfaceCounters . . . 479
2.487.3 InterfaceFastCounters . . . 479
2.488 show interfaces debounce | (debounce (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 480
2.488.1 InterfaceDebounceTime . . . 480
2.489 show interfaces description [...] . . . 480
2.489.1 InterfaceDescription . . . 480
2.490 show interfaces encapsulation dot1q vlan [...] . . . 480
2.490.1 InterfaceEncapsulations . . . 480
2.491 show interfaces encapsulation vlan [...] . . . 481
2.491.1 FlexibleEncapsulations . . . 481
2.492 show interfaces interactions | (interactions (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 482
2.492.1 InterfacesInteractionsDataBySpeed . . . 482
2.493 show interfaces mac [...] . . . 484
2.493.1 InterfaceMacData . . . 484
2.494 show interfaces queue length [...] . . . 485
2.494.1 QueueTypeQueueInfo . . . 485
2.494.2 SandQueueTypeQueueInfo . . . 485
2.495 show interfaces single INTF . . . 485
2.496 show interfaces status | (status (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 486

2.496.1 InterfaceStatusEntry . . . 486
2.497 show interfaces switchport configuration source [...] . . . 487
2.497.1 ConfigSource . . . 487
2.498 show interfaces switchport vlan mapping [...] . . . 488
2.498.1 InterfaceVlanMappings . . . 488
2.499 show interfaces switchport | (switchport (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 489
2.499.1 SwitchPort . . . 489
2.500 show interfaces transceiver [...] . . . 491
2.500.1 InterfaceTransceiverBase . . . 491
2.500.2 InterfaceTransceiver . . . 491
2.501 show interfaces transceiver diag test pattern [...] . . . 492
2.501.1 InterfacesTransceiverTestPatternConfiguration . . . 492
2.502 show interfaces transceiver diag test pattern capabilities [...] . . . 492
2.502.1 InterfacesTransceiverTestPatternCapabilities . . . 493
2.503 show interfaces transceiver diag test pattern counters [...] . . . 493
2.503.1 InterfacesTransceiverTestPatternCounters . . . 493
2.504 show interfaces transceiver dom [...] . . . 494
2.504.1 InterfaceTransceiverDomBase . . . 494
2.504.2 InterfaceTransceiverDom . . . 494
2.505 show interfaces transceiver eeprom [...] . . . 496
2.505.1 InterfacesTransceiverEepromIntf . . . 496
2.506 show interfaces transceiver hardware [...] . . . 498
2.506.1 InterfacesTransceiverHardware . . . 498
2.507 show interfaces transceiver performance-monitoring [...] . . . 499
2.507.1 InterfaceTransceiverPerformanceMonitoring . . . 499
2.508 show interfaces transceiver properties [...] . . . 499
2.508.1 TransceiverProperties . . . 499
2.509 show interfaces transceiver tuning [...] . . . 500
2.509.1 InterfaceTransceiverTuning . . . 500
2.510 show interfaces trunk | (trunk (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 501
2.510.1 InterfaceTrunk . . . 501
2.511 show interfaces unidirectional | (unidirectional (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...] . . . 502
2.511.1 InterfaceUnidirectional . . . 502
2.512 show interfaces vlans [...] . . . 502
2.512.1 InterfaceVlansModel . . . 502
2.513 show internet-exit policy [...] . . . 503
2.513.1 SfeIePolicyStatusModel . . . 503
2.514 show inventory . . . 504
2.514.1 CardSlot . . . 504
2.514.2 Clock . . . 505
2.514.3 FanTraySlot . . . 505
2.514.4 Fpga . . . 505
2.514.5 PowerSupplySlot . . . 505
2.514.6 StorageDevice . . . 505
2.514.7 SystemInformation . . . 505
2.514.8 XcvrSlot . . . 506
2.515 show ip . . . 506
2.515.1 IpMulticastRouting . . . 506
2.515.2 V6EcmpModel . . . 506

2.516 show ip access-lists [...] . . . 506
2.517 show ip address virtual source-nat sourceNatAddr . . . 506
2.518 show ip arp inspection interface [...] . . . 507
2.518.1 ArpInspectionInterfaceConfig . . . 507
2.519 show ip arp inspection statistics vlan | interface [...] . . . 508
2.519.1 ArpInspectionStatistics . . . 508
2.519.2 IntfArpInspectionStatistics . . . 508
2.519.3 VlanArpInspectionStatistics . . . 508
2.520 show ip arp inspection vlan [...] . . . 509
2.520.1 VlanArpInspectionStatus . . . 509
2.521 show ip as-path access-list NAME detail . . . 509
2.521.1 IpAsPathList . . . 509
2.522 show ip as-path access-list NAME summary . . . 509
2.523 show ip as-path access-list [...] . . . 510
2.524 show ip bgp ((community regex COMMREGEX as-path regex ASPREGEX) | (regex ASPREGEX community regex COMMREGEX)) [...] . . . 510
2.525 show ip bgp (IP | PREFIX) [...] . . . 510
2.526 show ip bgp (installed | not-installed) [...] . . . 510
2.527 show ip bgp [...] . . . 510
2.528 show ip bgp access-list AS_PATH_LIST_NAME [...] . . . 510
2.529 show ip bgp community regex REGEX [...] . . . 510
2.530 show ip bgp field-set FSNAME [...] . . . 511
2.531 show ip bgp large-community { LARGE_COMM_VALUE } detail exact [...] . . . 511
2.532 show ip bgp neighbors [...] . . . 512
2.532.1 BgpPeerList . . . 512
2.533 show ip bgp neighbors update errors [...] . . . 532
2.533.1 BgpNeighborUpdateError . . . 532
2.534 show ip bgp paths [...] . . . 532
2.534.1 BgpASPathList . . . 532
2.535 show ip bgp regex REGEX [...] . . . 533
2.536 show ip bgp summary [...] . . . 533
2.537 show ip community-list [...] . . . 533
2.537.1 IpCommunityList . . . 533
2.538 show ip decap-group [...] . . . 534
2.538.1 DecapGroupModelBase . . . 534
2.538.2 DecapGroupGreModel . . . 534
2.538.3 DecapGroupIpIpModel . . . 535
2.538.4 DecapGroupQosTcModel . . . 535
2.538.5 DecapGroupUdpModel . . . 535
2.539 show ip decap-group qos [...] . . . 535
2.540 show ip dhcp relay . . . 536
2.541 show ip dhcp snooping . . . 536
2.541.1 CircuitIdModel . . . 536
2.542 show ip dhcp snooping counters [...] . . . 537
2.542.1 VlanCounters . . . 537
2.543 show ip dhcp snooping counters debug . . . 537
2.543.1 SnoopingRelayCounters . . . 538
2.544 show ip dhcp snooping hardware . . . 538
2.544.1 VlanList . . . 538
2.545 show ip domain-name . . . 538
2.546 show ip extcommunity-list [...] . . . 538
2.546.1 IpExtCommunityList . . . 538
2.547 show ip hardware ale adj [...] . . . 539
2.547.1 AleL3AdjInfo . . . 539
2.548 show ip hardware ale adj acknowledgement [...] . . . 541

2.548.1 AcknowledgementFecEntry . . . 541
2.549 show ip hardware ale adj export [...] . . . 541
2.549.1 AleOpenConfigExportedFec . . . 541
2.550 show ip hardware ale adj proxy [...] . . . 542
2.550.1 FecProxyInfo . . . 542
2.551 show ip hardware ale counters egress bindings l2-adj [...] . . . 542
2.551.1 AleL2AdjCounterBindingsInfo . . . 542
2.552 show ip hardware ale counters egress bindings platform-key [...] . . . 542
2.552.1 AlePdckCounterBindingsInfo . . . 543
2.553 show ip hardware ale l2-adj [...] . . . 544
2.553.1 AleL2AdjInfo . . . 544
2.554 show ip hardware ale mac-address router gateway . . . 546
2.554.1 VlanListType . . . 546
2.555 show ip hardware ale routes [...] . . . 546
2.555.1 AleRouteInfo . . . 546
2.556 show ip hardware ale routes vrf all [...] . . . 546
2.557 show ip hardware ale vrf . . . 547
2.557.1 VrfStatus . . . 547
2.558 show ip hardware fib ecmp resilience [...] . . . 547
2.558.1 AleResEcmpRouteInfos . . . 547
2.559 show ip hardware fib ecmp resilience summary [...] . . . 548
2.560 show ip hardware fib load-balance distribution . . . 548
2.561 show ip hardware fib load-balance number [...] . . . 548
2.562 show ip hardware fib routes [...] . . . 548
2.563 show ip igmp access-list [...] . . . 549
2.564 show ip igmp groups GROUPADDR [...] . . . 549
2.564.1 IgmpGroup . . . 549
2.565 show ip igmp groups [...] . . . 549
2.566 show ip igmp groups count [...] . . . 549
2.567 show ip igmp groups interface INTF [...] . . . 550
2.568 show ip igmp groups interface INTF count [...] . . . 550
2.569 show ip igmp host-proxy (debug | interface) [...] . . . 551
2.569.1 IgmpHostProxyInterface . . . 551
2.570 show ip igmp host-proxy config-sanity . . . 552
2.570.1 IHPPotentialMisconfigs . . . 552
2.571 show ip igmp interface [...] . . . 553
2.571.1 IgmpInterface . . . 553
2.572 show ip igmp membership [...] . . . 553
2.572.1 IgmpMemGrpList . . . 553
2.573 show ip igmp profile [...] . . . 554
2.573.1 ProfileInfo . . . 554
2.574 show ip igmp snooping [...] . . . 555
2.574.1 IgmpSnoopingVlanInfo . . . 555
2.575 show ip igmp snooping counters [...] . . . 556
2.575.1 IgmpSnoopingCountersInterface . . . 556
2.576 show ip igmp snooping groups [...] . . . 556
2.576.1 VlanMembershipInfo . . . 556
2.577 show ip igmp snooping groups count [...] . . . 557
2.578 show ip igmp snooping groups detail [...] . . . 557
2.579 show ip igmp snooping groups proxy [...] . . . 557
2.579.1 IgmpSnoopingProxyPorts . . . 558
2.580 show ip igmp snooping mrouter [...] . . . 558
2.580.1 IgmpSnoopingMrouterVlan . . . 558
2.581 show ip igmp snooping querier [...] . . . 558
2.582 show ip igmp snooping querier counters [...] . . . 559

2.583 show ip igmp snooping querier membership [...] . . . 559
2.584 show ip igmp snooping querier membership vlan VLAN_ID group IPADDR . . . 559
2.585 show ip igmp snooping querier status [...] . . . 559
2.586 show ip igmp snooping report-flooding [...] . . . 559
2.586.1 IgmpSnoopingVlanReportFlooding . . . 559
2.587 show ip igmp static-groups [...] . . . 559
2.587.1 IgmpStaticGroupsInterface . . . 560
2.588 show ip igmp static-groups group [...] . . . 560
2.588.1 IgmpInterfacesList . . . 560
2.589 show ip igmp statistics [...] . . . 561
2.589.1 IgmpStatisticsInterface . . . 561
2.590 show ip interface [...] . . . 562
2.590.1 IpStatus . . . 562
2.591 show ip interface brief [...] . . . 563
2.591.1 IpStatusBrief . . . 563
2.592 show ip large-community-list [...] . . . 564
2.592.1 IpLargeCommunityList . . . 564
2.593 show ip load-sharing trident fields . . . 565
2.593.1 EcmpHashFieldsCommon . . . 565
2.593.2 EcmpHashFieldsGtp . . . 565
2.593.3 EcmpHashFieldsIp . . . 565
2.593.4 EcmpHashFieldsIpv6 . . . 565
2.594 show ip mfib PIM_MODE [...] . . . 566
2.594.1 MfibGroup . . . 566
2.595 show ip mfib [...] . . . 567
2.596 show ip mfib df [...] . . . 567
2.597 show ip mfib rpa [...] . . . 567
2.598 show ip mroute GROUP_OR_SOURCE1 interface [...] . . . 567
2.598.1 MrouteInterfaceDetails . . . 567
2.599 show ip mroute [...] . . . 569
2.599.1 Groups . . . 569
2.600 show ip mroute bidirectional GROUP_ADDR interface [...] . . . 570
2.601 show ip mroute bidirectional [...] . . . 570
2.602 show ip mroute bidirectional count [...] . . . 570
2.603 show ip mroute count [...] . . . 570
2.604 show ip mroute sparse-mode (sources | groups) [...] . . . 570
2.604.1 SourceGroupCollection . . . 570
2.605 show ip mroute vrf all [...] . . . 570
2.606 show ip msdp mesh-group [...] . . . 570
2.606.1 Group . . . 571
2.607 show ip msdp peer [...] . . . 572
2.607.1 PeerInfo . . . 572
2.608 show ip msdp pim sa-cache [...] . . . 572
2.609 show ip msdp rpf-peer RP [...] . . . 572
2.610 show ip msdp sa-cache [...] . . . 573
2.611 show ip msdp summary [...] . . . 573
2.611.1 PeerSummary . . . 573
2.612 show ip multicast boundary [...] . . . 573
2.612.1 McastBoundaryInterface . . . 573
2.613 show ip name-server . . . 573
2.614 show ip nat ((access-list | acl)) [...] . . . 574
2.614.1 NatAcl . . . 574
2.615 show ip nat counters [...] . . . 575
2.615.1 NatCounters . . . 575
2.616 show ip nat flow action [...] . . . 576

2.616.1 NatFlowActionPolicy . . . 576
2.617 show ip nat flow match [...] . . . 577
2.617.1 NatFlowMatchPolicy . . . 577
2.618 show ip nat flow profile [...] . . . 578
2.618.1 NatFlowProfilePolicy . . . 578
2.619 show ip nat logging . . . 578
2.620 show ip nat pool [...] . . . 579
2.620.1 NatPool . . . 579
2.621 show ip nat profile [...] . . . 579
2.621.1 NatProfile . . . 579
2.622 show ip nat service-list [...] . . . 579
2.622.1 NatServiceList . . . 580
2.623 show ip nat synchronization . . . 581
2.623.1 Version . . . 581
2.623.2 NatSyncFwdToPeer . . . 581
2.624 show ip nat synchronization advertised-translations [...] . . . 582
2.624.1 Translation . . . 582
2.625 show ip nat synchronization discovered-translations [...] . . . 583
2.626 show ip nat synchronization peer . . . 583
2.627 show ip nat translation (address-only | full-cone) flows [...] . . . 583
2.627.1 IntfTranslationFlows . . . 584
2.628 show ip nat translation DYNAMIC_TYPE [...] . . . 585
2.628.1 Translation . . . 585
2.629 show ip nat translation STATIC_TYPE [...] . . . 586
2.630 show ip nat translation [...] . . . 586
2.631 show ip nat translation address-only mapping [...] . . . 586
2.631.1 SrcIpTranslations . . . 586
2.632 show ip nat translation connection limit drop [...] . . . 587
2.632.1 DropTable . . . 587
2.633 show ip nat translation max-entries . . . 588
2.633.1 NatHostConnectionLimit . . . 588
2.633.2 NatPoolConnectionLimit . . . 589
2.634 show ip nat translation max-entries IPADDR . . . 589
2.635 show ip nat translation max-entries pool POOL . . . 589
2.636 show ip nat translation rates . . . 589
2.636.1 NatTranslationRates . . . 589
2.637 show ip nat translation summary [...] . . . 589
2.637.1 TranslationSummary . . . 590
2.638 show ip nat translation twice [...] . . . 591
2.638.1 Translation . . . 591
2.639 show ip nat translation vrf [...] . . . 591
2.639.1 Translation . . . 592
2.640 show ip ospf ((retransmission queue) | retransmission-list) [...] . . . 592
2.640.1 InstListOspfLsaRetransmissionList . . . 592
2.641 show ip ospf (request-list | (request queue)) [...] . . . 593
2.641.1 InstListOspfLsaRequestList . . . 593
2.642 show ip ospf [...] . . . 594
2.642.1 InstListOspfShowInstModel . . . 594
2.643 show ip ospf access-list [...] . . . 599
2.644 show ip ospf border-routers [...] . . . 599
2.644.1 InstListOspfBorderRoutersInstance . . . 599
2.645 show ip ospf counters [...] . . . 600
2.645.1 InstListOspfCountersInstance . . . 600
2.646 show ip ospf counters interface [...] . . . 601
2.646.1 InstListOspfCountersInterface . . . 601

2.647 show ip ospf counters interface rates [...] . . . 603
2.647.1 InstListOspfCountersInterfaceRates . . . 603
2.648 show ip ospf counters neighbor [...] . . . 605
2.648.1 InstListOspfCountersNeighbor . . . 605
2.649 show ip ospf database [...] . . . 606
2.649.1 InstListOspfDatabaseInstance . . . 606
2.650 show ip ospf database database-summary [...] . . . 610
2.650.1 InstListOspfDatabaseSummary . . . 610
2.651 show ip ospf interface [...] . . . 611
2.651.1 InstListOspfInterface . . . 611
2.652 show ip ospf lsa-log [...] . . . 612
2.652.1 InstListOspfLsaLogModel . . . 612
2.653 show ip ospf mpls ldp sync [...] . . . 613
2.653.1 InstListOspfMplsLdpSyncModel . . . 613
2.654 show ip ospf neighbor [...] . . . 614
2.654.1 InstListOspfNeighborsInstance . . . 614
2.655 show ip ospf neighbor adjacency-changes [...] . . . 615
2.655.1 InstListOspfNeighborAdjChanges . . . 616
2.656 show ip ospf neighbor summary [...] . . . 616
2.656.1 InstListOspfNeighborSummaryInstance . . . 616
2.657 show ip ospf sham-link [...] . . . 617
2.658 show ip ospf sham-link route [...] . . . 618
2.658.1 RibRoutesByProtocol . . . 618
2.659 show ip ospf spf-log [...] . . . 621
2.659.1 InstListOspfSpfLog . . . 621
2.660 show ip ospf summary [...] . . . 622
2.660.1 InstListOspfShowSummaryModel . . . 622
2.661 show ip pim access-list [...] . . . 623
2.662 show ip pim anycast-rp [...] . . . 623
2.663 show ip pim bidirectional df [...] . . . 623
2.663.1 Rpas . . . 623
2.664 show ip pim bidirectional rp [...] . . . 623
2.665 show ip pim bidirectional rp-hash GROUP_ADDR [...] . . . 623
2.666 show ip pim bidirectional upstream joins [...] . . . 624
2.667 show ip pim bsr [...] . . . 624
2.667.1 ElectedBsr . . . 624
2.668 show ip pim bsr counters [...] . . . 625
2.668.1 PimBsrErrorCounters . . . 625
2.669 show ip pim bsr debug . . . 626
2.669.1 PimBsrBorder . . . 626
2.669.2 PimBsrCandidate . . . 626
2.669.3 PimBsrRpCandidate . . . 627
2.669.4 PimBsrRpCandidateStatus . . . 627
2.669.5 ZoneAcl . . . 627
2.669.6 ZoneGroup . . . 627
2.670 show ip pim bsr rp [...] . . . 627
2.671 show ip pim bsr rp-hash GROUP [...] . . . 627
2.672 show ip pim config-sanity anycast-rp [...] . . . 627
2.672.1 RouteCheck . . . 628
2.673 show ip pim config-sanity bidirectional [...] . . . 629
2.673.1 MrtConfigCheck . . . 629
2.673.2 StaticRpSet . . . 629
2.674 show ip pim config-sanity igmp [...] . . . 630
2.674.1 InvalidBoundaryGroup . . . 630
2.674.2 NotDefinedBAcl . . . 630

2.674.3 NotStandardBAcl . . . 630
2.675 show ip pim config-sanity interface [...] . . . 630
2.676 show ip pim config-sanity neighbor [...] . . . 631
2.676.1 InvalidHello . . . 631
2.677 show ip pim config-sanity sparse-mode [...] . . . 631
2.678 show ip pim interface [...] . . . 632
2.678.1 PimInterface . . . 632
2.679 show ip pim neighbor [...] . . . 634
2.679.1 VrfPimNeighbors . . . 634
2.680 show ip pim neighbor bfd [...] . . . 634
2.680.1 VrfPimBfdNeighbors . . . 635
2.681 show ip pim protocol counters [...] . . . 635
2.681.1 PimMessageCounter . . . 635
2.681.2 PimMessageCountersDetails . . . 635
2.682 show ip pim register-source [...] . . . 635
2.683 show ip pim rp [...] . . . 635
2.684 show ip pim rp-candidate [...] . . . 636
2.684.1 PimRpCandidate . . . 636
2.685 show ip pim rp-hash GROUP_ADDR [...] . . . 636
2.686 show ip pim upstream joins [...] . . . 636
2.687 show ip prefix-list NAME detail . . . 636
2.687.1 IpPrefixList . . . 637
2.688 show ip prefix-list NAME summary . . . 637
2.689 show ip prefix-list [...] . . . 637
2.690 show ip rip access-list [...] . . . 637
2.691 show ip route (PREFIX | ADDR) longer-prefixes [...] . . . 638
2.691.1 IpRoutes . . . 638
2.692 show ip route [...] . . . 642
2.693 show ip route fec [...] . . . 643
2.693.1 FecRoutesModel . . . 643
2.694 show ip route fec protocol summary . . . 644
2.694.1 FecTypeBreakdown . . . 644
2.695 show ip route fec summary [...] . . . 644
2.695.1 FecBreakdown . . . 645
2.696 show ip route host [...] . . . 645
2.696.1 IpRoutesHost . . . 645
2.697 show ip route kernel unprogrammed [...] . . . 645
2.697.1 UnprogrammedRoutes . . . 646
2.698 show ip route multicast [...] . . . 646
2.698.1 MulticastRibRoute . . . 646
2.699 show ip route next-hop nexthop-group [...] . . . 646
2.700 show ip route next-hop vxlan [...] . . . 646
2.701 show ip route rpf unicast [...] . . . 646
2.701.1 UrpfInterface . . . 647
2.702 show ip route static not-installed bfd [...] . . . 647
2.702.1 NotInstalledBfd . . . 647
2.703 show ip route summary [...] . . . 648
2.703.1 IpRouteSummaryForVrf . . . 648
2.703.2 ServiceRoutingProtoModelStatus . . . 649
2.704 show ip route vrf all summary brief [...] . . . 649
2.705 show ip security applied-profile [...] . . . 649
2.705.1 IpsecAppliedProfile . . . 650
2.706 show ip security connection [...] . . . 651
2.706.1 IpsecConnection . . . 651
2.707 show ip security daemon [...] . . . 654

2.708 show ip security daemon log [...] . . . 655
2.709 show ip security fips status . . . 655
2.710 show ip security key controller local . . . 655
2.710.1 IpsecLocalKey . . . 655
2.711 show ip security key controller peer [...] . . . 655
2.711.1 IpsecPeer . . . 656
2.712 show ip security policy [...] . . . 656
2.712.1 IpsecPolicy . . . 656
2.713 show ip security profile [...] . . . 657
2.713.1 IpsecProfile . . . 657
2.714 show ip security security-association [...] . . . 657
2.714.1 IpsecSecurityAssociation . . . 657
2.715 show ip source binding static [...] . . . 657
2.715.1 IpMacBindingEntry . . . 658
2.716 show ip user route [...] . . . 658
2.716.1 IpRibRoutes . . . 658
2.717 show ip verify source . . . 658
2.717.1 IpsgInterfaceOperationalState . . . 659
2.718 show ip verify source detail . . . 659
2.718.1 IpsgBindingDetailStateList . . . 659
2.719 show ip verify source vlan . . . 659
2.719.1 IpsgVlanOperationalState . . . 659
2.720 show ip virtual-router [...] . . . 660
2.720.1 VarpRouterBase . . . 660
2.720.2 VarpRouterV4 . . . 660
2.720.3 VarpRouterV6 . . . 661
2.720.4 VirtualRouterMacModel . . . 661
2.720.5 MlagPeerRouterMac . . . 661
2.720.6 VarpRouterMac . . . 661
2.721 show ipv6 access-lists [...] . . . 661
2.722 show ipv6 address virtual source-nat sourceNatAddr . . . 662
2.723 show ipv6 bgp (IP | PREFIX) [...] . . . 662
2.724 show ipv6 bgp (installed | not-installed) [...] . . . 662
2.725 show ipv6 bgp [...] . . . 662
2.726 show ipv6 bgp access-list AS_PATH_LIST_NAME [...] . . . 662
2.727 show ipv6 bgp field-set FSNAME [...] . . . 662
2.728 show ipv6 bgp large-community { LARGE_COMM_VALUE } detail exact [...] . . . 662
2.729 show ipv6 bgp peers [...] . . . 663
2.730 show ipv6 bgp peers update errors [...] . . . 663
2.731 show ipv6 bgp regex REGEX [...] . . . 663
2.732 show ipv6 bgp summary [...] . . . 663
2.733 show ipv6 dhcp relay installed routes [...] . . . 663
2.733.1 Dhcp6InstalledRoutesPerVrf . . . 663
2.734 show ipv6 dhcp snooping . . . 664
2.735 show ipv6 dhcp snooping counters [...] . . . 664
2.736 show ipv6 dhcp snooping counters debug . . . 664
2.737 show ipv6 dhcp snooping hardware . . . 664
2.738 show ipv6 hardware ale routes [...] . . . 665
2.739 show ipv6 hardware ale routes vrf all [...] . . . 665
2.740 show ipv6 hardware fib ecmp resilience [...] . . . 665
2.741 show ipv6 hardware fib ecmp resilience summary [...] . . . 665
2.742 show ipv6 hardware fib routes [...] . . . 665
2.743 show ipv6 interface [...] . . . 666
2.743.1 Ip6Status . . . 666
2.744 show ipv6 interface brief [...] . . . 667

2.744.1 Ip6StatusBrief . . . 668
2.745 show ipv6 nd proxy [...] . . . 668
2.745.1 NdProxyPrefixInterface . . . 668
2.746 show ipv6 nd ra internal state [...] . . . 669
2.746.1 Interface . . . 669
2.746.2 VrfGlobalCounters . . . 670
2.747 show ipv6 nd ra neighbors [...] . . . 670
2.747.1 InterfaceNeighbors . . . 670
2.748 show ipv6 neighbor [...] . . . 671
2.748.1 IpV6Neighbor . . . 671
2.749 show ipv6 ospf [...] . . . 672
2.749.1 InstListOspf3ShowInstModel . . . 672
2.750 show ipv6 ospf access-list [...] . . . 675
2.751 show ipv6 ospf border-routers [...] . . . 675
2.751.1 InstListOspf3BorderRoutersInstance . . . 675
2.752 show ipv6 ospf database [...] . . . 676
2.752.1 InstListOspf3DatabaseInstance . . . 676
2.753 show ipv6 ospf database database-summary [...] . . . 679
2.753.1 InstListOspf3DatabaseSummary . . . 679
2.754 show ipv6 ospf interface [...] . . . 681
2.754.1 InstListOspf3Interface . . . 681
2.755 show ipv6 ospf lsa-log [...] . . . 682
2.755.1 InstListOspf3LsaLogInst . . . 682
2.756 show ipv6 ospf neighbor [...] . . . 684
2.756.1 InstListOspf3NeighborsInstance . . . 684
2.757 show ipv6 ospf request-list [...] . . . 685
2.757.1 InstListOspf3LsaRequestList . . . 686
2.758 show ipv6 ospf retransmission-list [...] . . . 686
2.758.1 InstListOspf3LsaRetransmissionList . . . 686
2.759 show ipv6 ospf spf-log [...] . . . 687
2.759.1 InstListOspf3SpfLogInst . . . 687
2.760 show ipv6 prefix-list NAME detail . . . 687
2.761 show ipv6 prefix-list NAME summary . . . 687
2.762 show ipv6 prefix-list [...] . . . 688
2.763 show ipv6 route (PREFIX | ADDR) longer-prefixes [...] . . . 689
2.763.1 Ip6Routes . . . 689
2.764 show ipv6 route [...] . . . 690
2.765 show ipv6 route fec [...] . . . 690
2.766 show ipv6 route fec summary [...] . . . 690
2.767 show ipv6 route host [...] . . . 690
2.767.1 Ip6RoutesHost . . . 690
2.768 show ipv6 route kernel unprogrammed [...] . . . 691
2.769 show ipv6 route multicast [...] . . . 691
2.770 show ipv6 route next-hop nexthop-group [...] . . . 691
2.771 show ipv6 route rpf unicast [...] . . . 691
2.771.1 Ip6UrpfInterface . . . 691
2.772 show ipv6 route static not-installed bfd [...] . . . 691
2.773 show ipv6 route summary [...] . . . 692
2.773.1 Ip6RouteSummaryForVrf . . . 692
2.774 show ipv6 route vrf all summary brief [...] . . . 693
2.775 show ipv6 user route [...] . . . 693
2.776 show ipv6 virtual-router [...] . . . 693
2.777 show isis (area proxy) | (INSTANCE area proxy) | (area proxy vrf VRFNAME) . . . 693
2.777.1 InstancesIsisShowAreaProxyModel . . . 693

2.778 show isis (database (detail | traffic-engineering)) | (INSTANCE database (detail | traffic-engineering)) | (database (detail | traffic-engineering) vrf VRFNAME) [...] 695
2.778.1 InstancesIsisLsdbDetailModel 695
2.779 show isis (dynamic flooding interfaces) | (INSTANCE dynamic flooding interfaces) | (dynamic flooding interfaces vrf VRFNAME) [...] 706
2.779.1 InstancesIsisDynFloodInterfacesModel 706
2.780 show isis (dynamic flooding nodes) | (INSTANCE dynamic flooding nodes) | (dynamic flooding nodes vrf VRFNAME) [...] 706
2.780.1 InstancesIsisDynFloodNodeModel 707
2.781 show isis (dynamic flooding paths) | (INSTANCE dynamic flooding paths) | (dynamic flooding paths vrf VRFNAME) [...] 707
2.781.1 InstancesIsisDynFloodPathsModel 707
2.782 show isis (dynamic flooding topology) | (INSTANCE dynamic flooding topology) | (dynamic flooding topology vrf VRFNAME) [...] 708
2.782.1 InstancesIsisDynFloodTopologyModel 708
2.783 show isis (flex-algo) | (INSTANCE flex-algo) | (flex-algo vrf VRFNAME) [...] 709
2.783.1 InstancesIsisShowFlexAlgoModel 709
2.784 show isis (graceful-restart) | (INSTANCE graceful-restart) | (graceful-restart vrf VRFNAME) 711
2.784.1 InstancesIsisGracefulRestartModel 711
2.785 show isis (local-convergence-delay) | (INSTANCE local-convergence-delay) | (local-convergence-delay vrf VRFNAME) [...] 713
2.785.1 InstancesIsisUloopModel 713
2.786 show isis (lsp log) | (INSTANCE lsp log) | (lsp log vrf VRFNAME) 714
2.786.1 InstancesIsisLspLogTableModel 714
2.787 show isis (lsp purges) | (INSTANCE lsp purges) | (lsp purges vrf VRFNAME) [...] 715
2.787.1 InstancesIsisLspPurgesModel 715
2.788 show isis (network topology) | (INSTANCE network topology) | (network topology vrf VRFNAME) [...] 716
2.788.1 InstancesIsisTopologyModel 716
2.789 show isis (segment-routing) | (INSTANCE segment-routing) | (segment-routing vrf VRFNAME) 718
2.789.1 InstancesSrSummaryModel 718
2.790 show isis (segment-routing adjacency-segments) | (INSTANCE segment-routing adjacency-segments) | (segment-routing adjacency-segments vrf VRFNAME) 719
2.790.1 InstancesAdjacencySegmentSummaryModel 719
2.791 show isis (segment-routing global-blocks) | (INSTANCE segment-routing global-blocks) | (segment-routing global-blocks vrf VRFNAME) 721
2.791.1 InstancesGlobalBlockSummaryModel 721
2.792 show isis (segment-routing prefix-segments) | (INSTANCE segment-routing prefix-segments) | (segment-routing prefix-segments vrf VRFNAME) [...] 722
2.792.1 InstancesPrefixSegmentSummaryModel 722
2.793 show isis (spf log) | (INSTANCE spf log) | (spf log vrf VRFNAME) [...] 723
2.793.1 InstancesIsisSpfLogTableModel 723
2.794 show isis (summary-address) | (INSTANCE summary-address) | (summary-address vrf VRFNAME) [...] 724
2.794.1 InstancesIsisLeakedRouteSumAddrInstanceModel 724
2.795 show isis counters | (INSTANCE counters) | (counters vrf VRFNAME) [...] 725
2.795.1 InstancesIsisInstanceCounters 725
2.796 show isis database | (INSTANCE database) | (database vrf VRFNAME) [...] 738
2.796.1 InstancesIsisLsdbSummaryModel 738
2.797 show isis flex-algo path [...] 740
2.797.1 FlexAlgoPathVrfModel 740
2.798 show isis hostname | (INSTANCE hostname) | (hostname vrf VRFNAME) 741
2.798.1 InstancesIsisHostnameInstanceModel 741
2.799 show isis interface | (INSTANCE interface) | (interface vrf VRFNAME) | (interface intf) [...]742

2.799.1 InstancesIsisInterfaceModel . . . 742
2.800 show isis mpls ldp sync [...] . . . 745
2.800.1 InstancesIsisMplsLdpSyncModel . . . 745
2.801 show isis neighbors | (INSTANCE neighbors) | (neighbors vrf VRFNAME) [...] . . . 746
2.801.1 InstancesIsisInstanceNeighbors . . . 746
2.802 show isis segment-routing tunnel [...] . . . 747
2.802.1 IsisSrTunnelTableEntry . . . 748
2.803 show isis summary | (INSTANCE summary) | (summary vrf VRFNAME) . . . 749
2.803.1 InstancesIsisInstanceSummaryModel . . . 749
2.804 show isis ti-lfa path [...] . . . 752
2.804.1 TilfaPathVrfModel . . . 752
2.805 show isis ti-lfa tunnel [...] . . . 753
2.805.1 TilfaTunnelTableEntry . . . 753
2.806 show isis topology | (INSTANCE topology) | (topology vrf VRFNAME) [...] . . . 754
2.807 show kernel interface addr [...] . . . 755
2.807.1 ShowKernelInterfaceAddrVrf . . . 755
2.808 show kernel interface counters [...] . . . 756
2.808.1 ShowKernelInterfaceCountersVrf . . . 756
2.809 show kernel ip acl [...] . . . 758
2.809.1 ShowKernelIpAclVrf . . . 758
2.810 show kernel ip arp [...] . . . 759
2.810.1 ShowKernelIpArpVrf . . . 759
2.811 show kernel ip counters [...] . . . 760
2.811.1 ShowKernelIpCountersVrfDetails . . . 760
2.812 show kernel ip interface addr [...] . . . 760
2.813 show kernel ip route [...] . . . 761
2.813.1 ShowKernelIpRouteVrf . . . 761
2.814 show kernel ipsec [...] . . . 762
2.814.1 ShowKernelIpsecConnections . . . 762
2.814.2 ShowKernelIpsecSecurityAssociations . . . 762
2.814.3 ShowKernelIpsecStrongSwan . . . 762
2.815 show kernel ipv6 acl [...] . . . 763
2.816 show kernel ipv6 counters [...] . . . 763
2.817 show kernel ipv6 interface addr [...] . . . 763
2.818 show kernel ipv6 neighbors [...] . . . 763
2.819 show kernel ipv6 route [...] . . . 763
2.820 show kernelFib rcf [...] . . . 763
2.821 show kernelFib scheduler [...] . . . 763
2.822 show l1 destination [...] . . . 763
2.822.1 L1Destination . . . 764
2.822.2 PhysicalIntfs . . . 764
2.822.3 VirtualIntfs . . . 764
2.823 show l1 matrix [...] . . . 764
2.823.1 L1MatrixEntry . . . 764
2.824 show l1 modules profile configuration [...] . . . 764
2.824.1 L1CardProfileEnumeration . . . 765
2.825 show l1 modules profile status [...] . . . 765
2.825.1 L1ProfileCardStatus . . . 765
2.826 show l1 path [...] . . . 765
2.826.1 L1Source . . . 765
2.827 show l1 source [...] . . . 765
2.828 show l1 source capabilities [...] . . . 766
2.828.1 SourceableIntfs . . . 766
2.829 show l2-protocol forwarding interfaces [...] . . . 766
2.829.1 L2ProtocolForwardingInterface . . . 766

2.830 show l2-protocol forwarding profile [...] . . . 766
2.830.1 L2ProtocolForwardingProfile . . . 766
2.831 show l2-protocol forwarding profile summary [...] . . . 767
2.831.1 L2ProtocolForwardingProfileSummary . . . 767
2.832 show l2Rib (input SOURCE) [...] . . . 768
2.832.1 HostEntry . . . 768
2.832.2 HostEntryAndSource . . . 769
2.833 show l2Rib (input SOURCE) floodset [...] . . . 769
2.833.1 FloodSetSummary . . . 769
2.834 show l2Rib output [...] . . . 769
2.835 show l2Rib output floodset [...] . . . 770
2.836 show l2Rib summary . . . 770
2.836.1 TableSummary . . . 770
2.837 show lacp aggregates [...] . . . 770
2.837.1 LacpMarkerMessages . . . 770
2.837.2 PortChannelAggregate . . . 770
2.838 show lacp counters [...] . . . 772
2.838.1 InterfaceLacpCounters . . . 772
2.838.2 PortChannelLacpCounters . . . 772
2.839 show lacp interface [...] . . . 773
2.839.1 InterfaceLacp . . . 773
2.839.2 PortChannelLacp . . . 774
2.840 show lacp interface counters [...] . . . 774
2.841 show lacp interface internal [...] . . . 775
2.841.1 InterfaceLacpInternal . . . 775
2.841.2 LacpSysIdBase . . . 776
2.841.3 MlagLacpSysId . . . 776
2.841.4 PortChannelLacpInternal . . . 776
2.842 show lacp interface neighbor [...] . . . 777
2.842.1 InterfaceLacpNeighbors . . . 777
2.842.2 PortChannelLacpNeighbors . . . 777
2.843 show lacp interface peer [...] . . . 777
2.844 show lacp internal [...] . . . 778
2.845 show lacp internal events [...] . . . 778
2.845.1 LacpInternalEventRecord . . . 778
2.846 show lacp neighbor [...] . . . 778
2.847 show lacp peer [...] . . . 779
2.848 show lacp sys-id [...] . . . 779
2.848.1 Details . . . 779
2.849 show led [...] . . . 779
2.849.1 LedSetting . . . 779
2.850 show led palette [...] . . . 779
2.850.1 LedCapability . . . 779
2.851 show led state [...] . . . 780
2.851.1 LedStateDict . . . 780
2.852 show led theme . . . 780
2.852.1 LightSettingTheme . . . 780
2.853 show led theme interface [...] . . . 780
2.854 show license [...] . . . 780
2.854.1 FeatureModel . . . 781
2.855 show license features . . . 781
2.856 show license files [...] . . . 781
2.857 show line system dom [...] . . . 781
2.858 show line system status [...] . . . 782
2.858.1 InterfacesTransceiverStatusValues . . . 782

2.859 show lldp [...] . . . 801
2.859.1 LldpInterface . . . 801
2.859.2 TlvType . . . 802
2.860 show lldp counters [...] . . . 802
2.860.1 LldpInterfaceTraffic . . . 802
2.861 show lldp local-info [...] . . . 803
2.861.1 LocalInterfaceInfo . . . 803
2.861.2 ManagementAddr . . . 808
2.861.3 SystemCapabilities . . . 808
2.862 show lldp neighbors [...] . . . 809
2.862.1 LldpNeighbor . . . 809
2.863 show lldp neighbors detail [...] . . . 810
2.863.1 LldpNeighborsList . . . 810
2.864 show lldp traffic [...] . . . 813
2.865 show load-balance destination port-channel PORT_CHANNEL { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-mac SRC_MAC) | (dst-mac DST_MAC) | (eth-type ETH_TYPE)) } | { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-ip-address SRC_IP) | (dst-ip-address DST_IP) | (ip-ttl IP_TTL) | (ip-protocol IP_PROTO)) | ((src-l4-port SRC_L4_PORT) | (dst-l4-port DST_L4_PORT)) } | { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-ipv6-address SRC_IPV6) | (dst-ipv6-address DST_IPV6) | (hop-limit HOP_LIMIT) | (next-header NEXT_HDR) | (flow-label FLOW_LABEL)) | ((src-l4-port SRC_L4_PORT) | (dst-l4-port DST_L4_PORT)) } . . . 814
2.866 show load-balance ecmp sand fields [...] . . . 815
2.866.1 Module . . . 815
2.867 show load-balance port-channel sand fields [...] . . . 818
2.867.1 Module . . . 818
2.867.2 AradProfile . . . 819
2.868 show load-balance profile [...] . . . 820
2.869 show load-balance server hardware connection history summary . . . 820
2.870 show load-balance server hardware errors . . . 820
2.870.1 LbPolicyErrors . . . 821
2.871 show load-balance server match-rule . . . 821
2.871.1 LoadBalancerMatchRule . . . 821
2.872 show load-balance server nexthop-group counters . . . 821
2.872.1 LbCounterEntry . . . 822
2.873 show load-balance server nexthop-group counters rates . . . 822
2.873.1 LbCounterRateEntry . . . 822
2.874 show load-balance server policy <policyName> match-rule <matchName> [...] . . . 822
2.875 show load-balance server policy [...] . . . 822
2.875.1 LoadBalancerVrfPolicy . . . 822
2.876 show load-balance server summary . . . 823
2.876.1 LoadBalancerVrfSummary . . . 823
2.877 show loop-protection [...] . . . 823
2.877.1 LoopProtectShowDetail . . . 823
2.878 show loop-protection counters . . . 824
2.878.1 LoopProtectCounter . . . 824
2.879 show loop-protection vlan VLANSET . . . 824
2.879.1 LoopProtectVlanIntfList . . . 824
2.880 show mac access-lists [...] . . . 825
2.881 show mac pause-frame . . . 825
2.882 show mac security counters [...] . . . 826
2.882.1 MacSecCountersInterface . . . 826
2.883 show mac security interface [...] . . . 828
2.883.1 MacsecInterface . . . 828
2.884 show mac security mka counters [...] . . . 830

2.884.1 MacsecMessageCountersInterface . . . 830
2.885 show mac security participants [...] . . . 831
2.885.1 MacsecParticipantsInterface . . . 831
2.886 show mac security profile [...] . . . 832
2.886.1 MacsecProfiles . . . 832
2.887 show mac security ptp bypass . . . 833
2.887.1 MacsecPtpBypassInterface . . . 833
2.888 show mac security status . . . 833
2.889 show mac source-check invalid . . . 834
2.890 show mac source-check system-mac . . . 834
2.891 show mac timestamp keyframe profile [...] . . . 834
2.891.1 Profile . . . 834
2.892 show mac traffic-policy field-set [...] . . . 834
2.892.1 MacTrafficPolicyMacFieldSetSource . . . 834
2.893 show mac traffic-policy summary . . . 835
2.893.1 MacTrafficPolicySourceSummary . . . 835
2.894 show maintenance . . . 835
2.894.1 MaintenanceProtocolInfo . . . 836
2.895 show maintenance bgp (IP_PEER | IP6_PEER | IP6_LL_PEER) [...] . . . 837
2.895.1 BgpPeerMaintenanceInfo . . . 837
2.896 show maintenance bgp ip all [...] . . . 838
2.896.1 BgpMaintenanceStatus . . . 838
2.897 show maintenance bgp ipv6 all [...] . . . 839
2.898 show maintenance bgp receiver route-map . . . 839
2.899 show maintenance debug [...] . . . 839
2.899.1 MaintenanceEventInfo . . . 839
2.900 show maintenance debug bgp . . . 839
2.901 show maintenance debug bgp (GROUP | IP_PEER | IP6_PEER | IP6_LL_PEER) [...] . . . 839
2.902 show maintenance groups . . . 840
2.902.1 BgpGroup . . . 840
2.902.2 InterfaceGroup . . . 840
2.903 show maintenance groups bgp [...] . . . 840
2.904 show maintenance groups interface [...] . . . 840
2.905 show maintenance interface [...] . . . 841
2.905.1 MaintenanceInterfaceBrief . . . 841
2.906 show maintenance interface detail [...] . . . 842
2.906.1 MaintenanceInterfaceStatus . . . 842
2.907 show maintenance profiles . . . 843
2.907.1 BgpProfile . . . 843
2.907.2 InterfaceProfile . . . 844
2.907.3 UnitProfile . . . 844
2.908 show maintenance profiles bgp [...] . . . 844
2.909 show maintenance profiles bgp default . . . 844
2.910 show maintenance profiles interface [...] . . . 844
2.911 show maintenance profiles interface default . . . 845
2.912 show maintenance profiles unit [...] . . . 845
2.913 show maintenance profiles unit default . . . 845
2.914 show maintenance stages [...] . . . 845
2.914.1 MaintenanceStage . . . 845
2.915 show maintenance summary . . . 845
2.915.1 MaintenanceRateMonSummary . . . 846
2.916 show maintenance units [...] . . . 846
2.916.1 MaintenanceUnit . . . 846
2.917 show management api gnmi . . . 847
2.917.1 GnmiEndpointStatus . . . 847

2.917.2 GrpctunnelClientStatus . . . 847
2.918 show management api gnmi access-list [...] . . . 848
2.919 show management api gribi . . . 849
2.919.1 GribiRpcStatus . . . 849
2.920 show management api gribi access-list [...] . . . 850
2.921 show management api http-commands . . . 851
2.921.1 HttpProtocolStatus . . . 851
2.921.2 SslProfile . . . 852
2.921.3 UnixProtocolStatus . . . 852
2.921.4 UserStatistics . . . 852
2.922 show management api http-commands ((ip access-list) | (ipv6 access-list)) [...] . . . 852
2.923 show management api http-commands https certificate . . . 852
2.924 show management api http-commands https cipher . . . 852
2.924.1 CipherSuite . . . 854
2.925 show management api http-commands https diffie-hellman . . . 855
2.926 show management api models . . . 856
2.926.1 IpfixOptions . . . 856
2.926.2 IsisOptions . . . 856
2.926.3 SflowOptions . . . 856
2.926.4 SysdbExcludesStatus . . . 857
2.927 show management api netconf . . . 857
2.928 show management api restconf . . . 857
2.929 show management api restconf access-list [...] . . . 857
2.930 show management archive . . . 858
2.931 show management cvx . . . 859
2.931.1 MgmtCvxClusterStatus . . . 859
2.932 show management cvx debug pingpong . . . 860
2.933 show management cvx mounts . . . 860
2.934 show management cvx service . . . 860
2.934.1 MgmtCvxService . . . 860
2.935 show management cvx service NAME . . . 860
2.936 show management dmf controller zerotouch . . . 860
2.937 show management dmf controller zerotouch detail . . . 861
2.937.1 LogMsg . . . 861
2.938 show management dmf dynamic configuration . . . 861
2.938.1 _Mode . . . 862
2.939 show management http-server . . . 862
2.939.1 HttpServiceVrf . . . 862
2.940 show management http-server ((ip access-list) | (ipv6 access-list)) [...] . . . 862
2.941 show management http-server counters . . . 862
2.941.1 VrfStats . . . 863
2.942 show management ldap . . . 863
2.942.1 LdapStatsModel . . . 863
2.942.2 LdapStatusModel . . . 863
2.942.3 ServerGroupModel . . . 864
2.943 show management package [...] . . . 864
2.944 show management package repository [...] . . . 864
2.944.1 RepositoryEntry . . . 864
2.945 show management security . . . 864
2.946 show management security session shared-secret profile NAME [...] . . . 866
2.946.1 SecretModel . . . 866
2.947 show management security session shared-secret profile [...] . . . 866
2.948 show management security ssl certificate [...] . . . 867
2.948.1 Certificate . . . 867
2.949 show management security ssl crl [...] . . . 868

2.949.1 Crl . . . 868
2.950 show management security ssl diffie-hellman [...] . . . 868
2.950.1 DiffieHellmanParameters . . . 868
2.951 show management security ssl key [...] . . . 868
2.952 show management security ssl profile [...] . . . 869
2.952.1 ProfileStatus . . . 869
2.953 show management security tpm pcr PCR_RANGE [...] . . . 869
2.954 show management security tpm pcr log [...] . . . 870
2.954.1 PcrLog . . . 870
2.955 show management ssh ((ip access-list) | (ipv6 access-list)) [...] . . . 870
2.956 show management ssh hostkey (KEYALGO | ecdsa) public . . . 870
2.957 show management ssh hostkey server cert [...] . . . 870
2.958 show management ssh trusted-ca key public [...] . . . 871
2.959 show management ssh user-keys revoke-list [...] . . . 871
2.960 show management telnet ((ip access-list) | (ipv6 access-list)) [...] . . . 871
2.961 show management telnet [...] . . . 871
2.962 show match-list prefix-ipv4 [...] . . . 871
2.962.1 MatchList . . . 871
2.963 show match-list prefix-ipv6 [...] . . . 871
2.964 show mdns counters . . . 871
2.965 show mdns links . . . 872
2.965.1 MdnsLinkList . . . 872
2.966 show mdns service name NAME [...] . . . 873
2.966.1 MdnsServiceInfo . . . 873
2.967 show mdns service rule NAME [...] . . . 873
2.967.1 MdnsServiceList . . . 874
2.968 show mdns service type [...] . . . 874
2.968.1 MdnsServiceTypeList . . . 874
2.969 show mdns status . . . 874
2.969.1 MdnsServerGatewayStatus . . . 875
2.970 show metawatch ptp . . . 876
2.970.1 PtpClockSummary . . . 876
2.970.2 PtpIntfVlanSummary . . . 877
2.971 show mlag [...] . . . 878
2.971.1 Detail . . . 879
2.972 show mlag config-sanity [...] . . . 880
2.972.1 GlobalFeatureParam . . . 880
2.972.2 IntfFeatureParam . . . 881
2.973 show mlag interfaces [...] . . . 882
2.973.1 MlagInterface . . . 882
2.974 show mlag interfaces interlock filter egress [...] . . . 883
2.975 show mlag interfaces members [...] . . . 883
2.976 show mlag issu compatibility URL . . . 883
2.977 show mlag issu warnings . . . 883
2.978 show mlag tunnel [...] . . . 883
2.978.1 PktTypeCounters . . . 883
2.979 show mld membership [...] . . . 883
2.979.1 MldMembershipInterface . . . 884
2.980 show mld querier [...] . . . 884
2.980.1 MldQuerierInterface . . . 884
2.981 show mld snooping [...] . . . 885
2.981.1 MldSnoopingVlanInfo . . . 885
2.982 show mld snooping counters [...] . . . 886
2.982.1 MldSnoopingCountersInterface . . . 886
2.983 show mld snooping groups [...] . . . 886

2.984 show mld snooping groups count [...] . . . 886
2.985 show mld snooping groups detail [...] . . . 886
2.985.1 VlanMembershipDetailInfo . . . 887
2.986 show mld snooping mrouter [...] . . . 887
2.987 show mld snooping querier [...] . . . 887
2.987.1 MldSnoopingVlanQuerier . . . 888
2.988 show mld snooping querier counters [...] . . . 888
2.988.1 MldSnoopingPacketCounters . . . 888
2.989 show mld snooping querier detail [...] . . . 888
2.990 show mld snooping querier membership [...] . . . 889
2.990.1 VlanMembership . . . 889
2.991 show mld static-group access-list ACL_NAME [...] . . . 889
2.991.1 RuleInfo . . . 889
2.992 show mld statistics [...] . . . 890
2.992.1 MldStatisticsInterface . . . 890
2.993 show mld summary [...] . . . 890
2.993.1 MldSummaryInterface . . . 891
2.994 show module [...] . . . 891
2.994.1 Module . . . 891
2.995 show module power [...] . . . 893
2.995.1 Module . . . 893
2.996 show monitor connectivity [...] . . . 895
2.996.1 ConnMonitorVrf . . . 895
2.997 show monitor device-health . . . 896
2.997.1 HealthCategoryModel . . . 896
2.998 show monitor ethernet reflectors [...] . . . 896
2.998.1 ReflectorInterfaceModel . . . 896
2.999 show monitor hadoop . . . 897
2.999.1 BasicJobInfo . . . 897
2.999.2 JobInfo . . . 897
2.1000 show monitor hadoop cluster CLUSTER_NAME counters . . . 899
2.1001 show monitor hadoop cluster CLUSTER_NAME history . . . 899
2.1001.1 JobHistoryData . . . 899
2.1002 show monitor hadoop cluster CLUSTER_NAME history job JOB . . . 899
2.1003 show monitor hadoop cluster CLUSTER_NAME jobs . . . 899
2.1004 show monitor hadoop cluster CLUSTER_NAME jobs JOB . . . 899
2.1005 show monitor hadoop cluster CLUSTER_NAME jobs JOB counters . . . 900
2.1006 show monitor hadoop cluster CLUSTER_NAME node [...] . . . 901
2.1006.1 TaskTracker . . . 901
2.1007 show monitor hadoop cluster CLUSTER_NAME status . . . 903
2.1007.1 ClusterError . . . 903
2.1007.2 ClusterInfo . . . 903
2.1007.3 TaskTrackers . . . 904
2.1007.4 InterfaceTaskTrackers . . . 904
2.1008 show monitor hadoop cluster all . . . 905
2.1009 show monitor hadoop counters . . . 905
2.1010 show monitor hadoop history . . . 905
2.1011 show monitor hadoop node (host HOST) | (interface INTF) counters . . . 905
2.1012 show monitor hadoop node (host HOST) | (interface INTF) jobs [...] . . . 905
2.1013 show monitor hadoop node (host HOST) | (interface INTF) running-tasks [...] . . . 906
2.1014 show monitor hadoop node (host HOST) | (interface INTF) running-tasks cluster CLUSTER_NAME job JOB task TASK . . . 906
2.1015 show monitor hadoop node all [...] . . . 906
2.1016 show monitor hadoop node all counters . . . 906
2.1016.1 TaskTrackerCounter . . . 906

2.1017 show monitor hadoop node host HOST status . . . 906
2.1018 show monitor hadoop node interface INTF status . . . 906
2.1019 show monitor hadoop status . . . 907
2.1020 show monitor hadoop traffic burst [...] . . . 907
2.1020.1 InterfaceBurst . . . 907
2.1021 show monitor link qualification generators [...] . . . 908
2.1021.1 LinkQualificationGeneratorIntfModel . . . 908
2.1022 show monitor link qualification generators profiles [...] . . . 908
2.1022.1 LinkQualificationProfileModel . . . 909
2.1023 show monitor rfc2544 generators profiles [...] . . . 909
2.1023.1 Rfc2544ProfileModel . . . 909
2.1024 show monitor rfc2544 generators test-report [...] . . . 910
2.1024.1 IntfTestReportModel . . . 910
2.1025 show monitor security awake [...] . . . 915
2.1025.1 MsaNucleus . . . 915
2.1025.2 MsaStatus . . . 915
2.1026 show monitor security awake counters [...] . . . 917
2.1026.1 FlowCounters . . . 917
2.1026.2 IpfixCounters . . . 917
2.1026.3 NucleusCounters . . . 918
2.1027 show monitor security awake dpi memory . . . 918
2.1027.1 MsaDpiMemoryDetails . . . 919
2.1028 show monitor security awake dpi memory allocations [...] . . . 919
2.1028.1 MsaDpiCtxMemoryAllocations . . . 919
2.1029 show monitor security awake dpi memory pools . . . 919
2.1029.1 MsaDpiMemoryPoolDetails . . . 919
2.1030 show monitor security awake flow-table [...] . . . 921
2.1030.1 FlowGroup . . . 921
2.1031 show monitor server PROTOCOL [...] . . . 923
2.1031.1 ServerProbeStats . . . 923
2.1032 show monitor server-failure . . . 923
2.1033 show monitor server-failure history . . . 923
2.1033.1 Server . . . 924
2.1034 show monitor server-failure servers IP_ADDR . . . 924
2.1035 show monitor server-failure servers [...] . . . 924
2.1036 show monitor session [...] . . . 925
2.1036.1 MirrorSession . . . 925
2.1037 show monitor system . . . 928
2.1038 show monitor telemetry inband . . . 929
2.1038.1 InbandTelemetryProfiles . . . 929
2.1038.2 SamplePolicyModel . . . 929
2.1039 show monitor telemetry inband profile [...] . . . 938
2.1040 show monitor telemetry inband profile summary [...] . . . 938
2.1040.1 IntfList . . . 938
2.1041 show monitor telemetry influx . . . 938
2.1041.1 DestinationInfluxdb . . . 938
2.1041.2 SourceSocket . . . 938
2.1042 show monitor telemetry path characteristics [...] . . . 939
2.1042.1 ItsPeer . . . 939
2.1043 show monitor telemetry path counters [...] . . . 940
2.1044 show monitor telemetry postcard counters . . . 940
2.1045 show monitor telemetry postcard policy . . . 940
2.1045.1 Collection . . . 940
2.1045.2 SampleTcpUdpChecksum . . . 941
2.1046 show monitor telemetry postcard policy profile [...] . . . 941

2.1046.1 ProfileModel . . . 941
2.1047 show monitor telemetry postcard sample policy [...] . . . 941
2.1047.1 SamplePolicyModel . . . 941
2.1048 show mpls bgp labeled-unicast bindings [...] . . . 942
2.1048.1 MplsLabelBinding . . . 942
2.1048.2 MplsLdpBindingAdvertisements . . . 943
2.1049 show mpls config route [...] . . . 944
2.1049.1 MplsRoute . . . 944
2.1050 show mpls hardware ale adj [...] . . . 948
2.1050.1 MplsFecRequestTableEntry . . . 948
2.1051 show mpls interface [...] . . . 948
2.1051.1 MplsInterfaceInfo . . . 949
2.1051.2 MplsInterfaceIngressCounters . . . 949
2.1052 show mpls interface counters [...] . . . 949
2.1053 show mpls label ranges . . . 949
2.1053.1 MplsLabelRange . . . 949
2.1054 show mpls ldp [...] . . . 950
2.1054.1 MplsLdpStatus . . . 950
2.1055 show mpls ldp access-list [...] . . . 952
2.1056 show mpls ldp bindings [...] . . . 952
2.1057 show mpls ldp bindings mldp [...] . . . 952
2.1058 show mpls ldp discovery [...] . . . 952
2.1058.1 MplsLdpDiscovery . . . 952
2.1059 show mpls ldp neighbor [...] . . . 954
2.1059.1 MplsLdpNeighborList . . . 954
2.1060 show mpls ldp rcf [...] . . . 961
2.1061 show mpls ldp tunnel [...] . . . 961
2.1061.1 LdpTunnelTableEntry . . . 961
2.1062 show mpls ldp tunnel mldp [...] . . . 961
2.1062.1 MldpTunnelTableEntry . . . 961
2.1063 show mpls lfib counters [...] . . . 961
2.1063.1 MplsLfibCounterLabelEntry . . . 962
2.1064 show mpls lfib counters no-resource . . . 962
2.1065 show mpls lfib counters summary . . . 962
2.1066 show mpls lfib input bgp labeled-unicast . . . 962
2.1067 show mpls lfib route [...] . . . 963
2.1068 show mpls parsing speculative . . . 963
2.1069 show mpls ping access-list [...] . . . 963
2.1070 show mpls route [...] . . . 964
2.1070.1 MplsHwRoute . . . 964
2.1071 show mpls route summary . . . 966
2.1072 show mpls rsvp . . . 967
2.1072.1 GrHelper . . . 968
2.1072.2 GrSpeaker . . . 968
2.1072.3 HitlessRestart . . . 968
2.1072.4 LspCount . . . 968
2.1073 show mpls rsvp ((ip access-list) | (ipv6 access-list)) [...] . . . 968
2.1074 show mpls rsvp bandwidth [...] . . . 968
2.1074.1 RsvpBandwidthIntfModel . . . 968
2.1075 show mpls rsvp bandwidth summary [...] . . . 969
2.1075.1 RsvpBandwidthIntfSummaryModel . . . 969
2.1076 show mpls rsvp counters [...] . . . 969
2.1076.1 RsvpErrorCounter . . . 969
2.1076.2 RsvpInterfaceCounter . . . 969
2.1077 show mpls rsvp neighbor [...] . . . 970

2.1077.1 RsvpNeighbor . . . 970
2.1078 show mpls rsvp neighbor summary . . . 971
2.1078.1 RsvpNeighborStatistics . . . 971
2.1079 show mpls rsvp session [...] . . . 972
2.1079.1 RsvpSession . . . 972
2.1080 show mpls segment-routing bindings [...] . . . 974
2.1081 show mpls tunnel counters [...] . . . 974
2.1081.1 MplsTunnelCountersEntry . . . 974
2.1082 show mpls tunnel fib . . . 975
2.1082.1 MplsTunnelFibEntry . . . 975
2.1083 show mpls tunnel static [...] . . . 976
2.1083.1 StaticTunnelTableEntry . . . 976
2.1084 show mpls tunnel termination qos maps [...] . . . 976
2.1085 show msdp mesh-group [...] . . . 976
2.1086 show msdp rpf-peer RP [...] . . . 976
2.1087 show msrp [...] . . . 977
2.1087.1 Intf . . . 977
2.1087.2 SrClassGlobal . . . 977
2.1088 show msrp counters [...] . . . 978
2.1088.1 Intf . . . 978
2.1089 show msrp interfaces (streams | listeners | talkers) [...] . . . 978
2.1089.1 Intf . . . 978
2.1090 show msrp mrp [...] . . . 979
2.1090.1 IntfDatabase . . . 979
2.1091 show msrp streams [...] . . . 980
2.1091.1 Stream . . . 980
2.1092 show msrp streams count . . . 981
2.1093 show msrp streams propagation [...] . . . 982
2.1093.1 Stream . . . 982
2.1094 show multicast (ipv4 | ipv6) evpn decap joined [...] . . . 983
2.1094.1 MrouteVxlanDecapInterface . . . 983
2.1095 show multicast (ipv4 | ipv6) source route export [...] . . . 983
2.1096 show multicast AF evpn decap received . . . 983
2.1096.1 MulticastTunnelVlan . . . 983
2.1096.2 UnderlayReplicationModel . . . 984
2.1097 show multicast AF evpn decap received flood . . . 984
2.1097.1 MulticastFloodVlan . . . 984
2.1098 show multicast AF evpn encap . . . 984
2.1099 show multicast AF evpn encap flood . . . 984
2.1100 show multicast fib ipv4 PIM_MODE [...] . . . 984
2.1101 show multicast fib ipv4 [...] . . . 984
2.1102 show multicast fib ipv4 df [...] . . . 985
2.1103 show multicast fib ipv4 rpa [...] . . . 985
2.1104 show multicast fib ipv6 PIM_MODE [...] . . . 985
2.1105 show multicast fib ipv6 [...] . . . 985
2.1106 show multicast ipv4 boundary [...] . . . 985
2.1107 show multicast ipv4 evpn encap mapping (vrf VRF) OVERLAY_GRP . . . 985
2.1108 show multicast ipv6 boundary [...] . . . 985
2.1109 show mvpn ipv4 vrf . . . 985
2.1109.1 MvpnVrfPmsis . . . 985
2.1110 show mvrp [...] . . . 986
2.1110.1 Intf . . . 986
2.1111 show mvrp mrp [...] . . . 986
2.1111.1 IntfDatabase . . . 986
2.1112 show mvrp vlans [...] . . . 987

2.1112.1 Vlan . . . 987
2.1113 show nat map-t domain [...] . . . 987
2.1113.1 Domain . . . 987
2.1114 show nat map-t domain counters [...] . . . 988
2.1114.1 DomainCountersSummary . . . 988
2.1115 show nat map-t domain counters drop [...] . . . 988
2.1115.1 DomainDropCounters . . . 988
2.1116 show nat map-t domain counters exception [...] . . . 988
2.1116.1 DomainExceptionCounters . . . 988
2.1117 show nat map-t domain counters offload [...] . . . 989
2.1117.1 DomainOffloadCounters . . . 989
2.1118 show nat map-t domain counters rule [...] . . . 989
2.1118.1 DomainRuleCounters . . . 989
2.1119 show nat map-t summary . . . 990
2.1119.1 DomainSummary . . . 990
2.1120 show neighbor classification [...] . . . 990
2.1120.1 NbrClassificationModel . . . 990
2.1121 show neighbor classification summary [...] . . . 990
2.1121.1 NbrClassificationSummaryModel . . . 990
2.1122 show network physical-topology hosts [...] . . . 990
2.1122.1 Host . . . 991
2.1123 show network physical-topology neighbors [...] . . . 991
2.1123.1 DirectedEdge . . . 991
2.1124 show nexthop-group [...] . . . 992
2.1124.1 NexthopGroup . . . 992
2.1125 show nexthop-group id [...] . . . 993
2.1126 show nexthop-group id mapping . . . 993
2.1127 show nexthop-group summary . . . 994
2.1128 show nexthop-group tunnel [...] . . . 994
2.1128.1 NexthopGroupTunnelTableEntry . . . 994
2.1129 show ntp ((ip access-list) | (ipv6 access-list)) [...] . . . 994
2.1130 show ntp associations . . . 994
2.1130.1 NtpPeer . . . 994
2.1131 show ntp status . . . 995
2.1132 show openflow . . . 996
2.1132.1 BindModeEntries . . . 996
2.1132.2 ConnectedController . . . 996
2.1132.3 ControllerConfig . . . 997
2.1132.4 RvlanEntries . . . 997
2.1133 show openflow flows [...] . . . 997
2.1134 show openflow groups [...] . . . 997
2.1134.1 Group . . . 997
2.1135 show openflow ports . . . 998
2.1135.1 Entry . . . 998
2.1136 show openflow profiles . . . 999
2.1136.1 OpenFlowTableProfile . . . 999
2.1137 show openflow queues . . . 1000
2.1137.1 OpenFlowQStats . . . 1000
2.1137.2 PortQ . . . 1000
2.1138 show openflow statistics . . . 1000
2.1138.1 Stats . . . 1000
2.1139 show openstack (ip | ipv6) access-list [...] . . . 1001
2.1140 show openstack agent uuid . . . 1001
2.1141 show openstack config all-regions . . . 1002
2.1141.1 RegionConfig . . . 1002

2.1142 show openstack config networks vni mapping . . . 1003
2.1142.1 VlanToVniMap . . . 1004
2.1143 show openstack config region REGION_NAME . . . 1004
2.1144 show openstack config region REGION_NAME timestamp . . . 1004
2.1145 show openstack features . . . 1004
2.1146 show openstack grace-period . . . 1004
2.1147 show openstack instances [...] . . . 1005
2.1147.1 Region . . . 1005
2.1148 show openstack networks [...] . . . 1007
2.1149 show openstack physical-network [...] . . . 1008
2.1149.1 Switch . . . 1008
2.1150 show openstack regions [...] . . . 1008
2.1151 show openstack resource-pool vlan region REGION_NAME uuid . . . 1008
2.1152 show openstack resource-pools [...] . . . 1008
2.1152.1 PhysicalNetworkVlanPool . . . 1009
2.1153 show openstack segments { (host HOSTNAME) | (network NETWORK_NAME_OR_ID) | (port PORT_NAME_OR_ID) | (instance INSTANCE_NAME_OR_ID) | (region REGION_NAME) | (switch SWITCH) | (tenant TENANT_NAME_OR_ID) } . . . 1009
2.1153.1 RegionInstanceBinding . . . 1009
2.1154 show openstack sync-timeout . . . 1010
2.1155 show openstack tenants [...] . . . 1010
2.1156 show openstack vlans . . . 1010
2.1157 show openstack vms [...] . . . 1010
2.1158 show ospfv3 [...] . . . 1011
2.1158.1 AddressFamilyOspf3MultiAfShowInstModel . . . 1011
2.1159 show ospfv3 access-list [...] . . . 1012
2.1160 show ospfv3 border-routers [...] . . . 1012
2.1160.1 AddressFamilyOspf3MultiAfBorderRoutersInstance . . . 1013
2.1161 show ospfv3 counters [...] . . . 1014
2.1161.1 AddressFamilyOspf3CountersInstance . . . 1014
2.1162 show ospfv3 counters interface [...] . . . 1014
2.1162.1 AddressFamilyOspf3CountersInterface . . . 1014
2.1163 show ospfv3 counters neighbor [...] . . . 1016
2.1163.1 AddressFamilyOspf3MultiAfCountersNeighor . . . 1016
2.1164 show ospfv3 database [...] . . . 1017
2.1164.1 AddressFamilyOspf3MultiAfDatabaseInstance . . . 1017
2.1165 show ospfv3 database database-summary [...] . . . 1018
2.1165.1 AddressFamilyOspf3MultiAfDatabaseSummary . . . 1018
2.1166 show ospfv3 interface [...] . . . 1018
2.1166.1 AddressFamilyOspf3MultiAfInterface . . . 1019
2.1167 show ospfv3 lsa-log [...] . . . 1019
2.1167.1 AddressFamilyOspf3MultiAfLsaLogInst . . . 1019
2.1168 show ospfv3 neighbor [...] . . . 1019
2.1168.1 AddressFamilyOspf3MultiAfNeighborsInstance . . . 1019
2.1169 show ospfv3 request-list [...] . . . 1020
2.1169.1 AddressFamilyOspf3MultiAfLsaRequestList . . . 1020
2.1170 show ospfv3 retransmission-list [...] . . . 1020
2.1170.1 AddressFamilyOspf3MultiAfLsaRetransmissionList . . . 1020
2.1171 show ospfv3 spf-log [...] . . . 1020
2.1171.1 AddressFamilyOspf3MultiAfSpfLogInst . . . 1021
2.1172 show p4-runtime . . . 1021
2.1172.1 P4RtDeviceStatusEntry . . . 1021
2.1172.2 P4RtTransport . . . 1021
2.1173 show p4-runtime device . . . 1022
2.1173.1 P4RtDeviceStatusEntryDetail . . . 1022

2.1174 show p4-runtime device counters packet-io . . . 1022
2.1174.1 P4RtDeviceCounterEntry . . . 1022
2.1175 show patch panel [...] . . . 1023
2.1175.1 Patch . . . 1023
2.1176 show patch panel forwarding . . . 1029
2.1176.1 LabelForwardingInfoTable . . . 1029
2.1176.2 PortForwardingInfoTable . . . 1031
2.1176.3 TaggedForwardingInfoTable . . . 1032
2.1177 show patch panel summary . . . 1033
2.1178 show path-selection application counters [...] . . . 1033
2.1178.1 DpsLbProfiles . . . 1033
2.1179 show path-selection load-balance counters [...] . . . 1034
2.1180 show path-selection mtu [...] . . . 1035
2.1180.1 DpsPeer . . . 1035
2.1181 show path-selection paths [...] . . . 1036
2.1182 show path-selection paths export . . . 1036
2.1182.1 DpsExport . . . 1037
2.1183 show pci . . . 1037
2.1183.1 PciErrorSum . . . 1037
2.1184 show pci detail . . . 1037
2.1184.1 PciDetailError . . . 1037
2.1185 show pci tree . . . 1037
2.1186 show peer-filter [...] . . . 1038
2.1186.1 PeerFilter . . . 1038
2.1187 show pim (ipv4 | ipv6) anycast-rp [...] . . . 1038
2.1188 show pim (ipv4 | ipv6) bsr [...] . . . 1038
2.1189 show pim (ipv4 | ipv6) bsr counters [...] . . . 1038
2.1190 show pim (ipv4 | ipv6) bsr debug . . . 1038
2.1191 show pim (ipv4 | ipv6) bsr rp [...] . . . 1039
2.1192 show pim (ipv4 | ipv6) bsr rp-hash GROUP [...] . . . 1039
2.1193 show pim (ipv4 | ipv6) config-sanity anycast-rp [...] . . . 1039
2.1193.1 GenRouteCheck . . . 1039
2.1194 show pim (ipv4 | ipv6) interface [...] . . . 1039
2.1195 show pim (ipv4 | ipv6) neighbor [...] . . . 1039
2.1196 show pim (ipv4 | ipv6) neighbor bfd [...] . . . 1039
2.1197 show pim (ipv4 | ipv6) register-source [...] . . . 1040
2.1198 show pim (ipv4 | ipv6) rp [...] . . . 1040
2.1199 show pim (ipv4 | ipv6) rp-candidate [...] . . . 1040
2.1200 show pim evpn gateway dr [...] . . . 1040
2.1200.1 EvpnGatewayDr . . . 1040
2.1201 show pim ipv4 sparse-mode non-dr drop-rules [...] . . . 1040
2.1201.1 NonDrDropRuleInterface . . . 1041
2.1202 show pim ipv6 sparse-mode join [...] . . . 1041
2.1203 show pim ipv6 sparse-mode route IP1 IP2 interface [...] . . . 1041
2.1204 show pim ipv6 sparse-mode route [...] . . . 1041
2.1205 show pim ipv6 sparse-mode route count [...] . . . 1041
2.1206 show pim ipv6 sparse-mode rp [...] . . . 1041
2.1207 show pim ipv6 sparse-mode rp hash GROUP [...] . . . 1042
2.1208 show pim protocol counters [...] . . . 1042
2.1209 show pim sparse-mode evpn cli membership [...] . . . 1042
2.1209.1 VlanMembership . . . 1042
2.1210 show pim sparse-mode evpn gateway [...] . . . 1042
2.1210.1 EvpnGatewayVrf . . . 1042
2.1211 show pim sparse-mode evpn pim membership [...] . . . 1042
2.1212 show pim sparse-mode pim evpn membership [...] . . . 1043

2.1213 show platform (fap | arad) eci hardware-sync [...] . . . 1043
2.1213.1 AradEciTimeSyncInfo . . . 1043
2.1214 show platform (fap | arad) zuma [...] . . . 1043
2.1214.1 ZumaMapping . . . 1043
2.1215 show platform (fap | arad) zuma detail [...] . . . 1043
2.1215.1 ZumaDetails . . . 1043
2.1216 show platform (fap | arad | jericho | jericho2) [...] . . . 1044
2.1216.1 FapInfo . . . 1044
2.1217 show platform (fap | arad | jericho | jericho2) cint [...] . . . 1044
2.1217.1 CintFap . . . 1044
2.1218 show platform (fap | arad | jericho | jericho2) copp mapping [...] . . . 1044
2.1218.1 AradGenericCoppClassStats . . . 1044
2.1218.2 AradPolicerCoppClassStats . . . 1045
2.1219 show platform (fap | arad | jericho | jericho2) eedb outac [...] . . . 1046
2.1219.1 EedbOutAcPoolMap . . . 1046
2.1220 show platform (fap | arad | jericho | jericho2) evpn mpls decap [...] . . . 1047
2.1220.1 L2EvpnMplsDecapMapping . . . 1047
2.1220.2 L3EvpnMplsDecapMapping . . . 1047
2.1221 show platform (fap | arad | jericho | jericho2) evpn mpls encap [...] . . . 1048
2.1221.1 VlanEvpnMplsEncapMap . . . 1048
2.1222 show platform (fap | arad | jericho | jericho2) fabric load distribution [...] . . . 1049
2.1222.1 WfdConfigNumSerdes . . . 1049
2.1223 show platform (fap | arad | jericho | jericho2) fabric load distribution destination (fap | arad | jericho | jericho2) [...] . . . 1049
2.1223.1 FapWfdSerdesWeights . . . 1049
2.1224 show platform (fap | arad | jericho | jericho2) flow tracking sampled flow-table hw [...] . . . 1049
2.1224.1 FlowTableFapLevelInfo . . . 1049
2.1225 show platform (fap | arad | jericho | jericho2) ip route prefix compression tcam [...] . . . 1050
2.1225.1 PrefixCompressionTcamEntries . . . 1050
2.1226 show platform (fap | arad | jericho | jericho2) label sharing inlif shadow [...] . . . 1051
2.1226.1 LabelSharingInLifEntriesForFap . . . 1051
2.1227 show platform (fap | arad | jericho | jericho2) label sharing isem (shadow | hardware | diff) [...] 1055
2.1227.1 LabelSharingIsemForFap . . . 1055
2.1228 show platform (fap | arad | jericho | jericho2) label sharing ppmc l2 shadow [...] . . . 1056
2.1228.1 PpmcLabelSharingEntries . . . 1056
2.1229 show platform (fap | arad | jericho | jericho2) label sharing ppmc l3 shadow [...] . . . 1056
2.1230 show platform (fap | arad | jericho | jericho2) lag [...] . . . 1056
2.1230.1 AradLags . . . 1057
2.1231 show platform (fap | arad | jericho | jericho2) mapping [...] . . . 1058
2.1231.1 PortMappings . . . 1058
2.1232 show platform (fap | arad | jericho | jericho2) mroute [...] . . . 1059
2.1232.1 FapToIpv4MroutesMapping . . . 1059
2.1232.2 FapToIpv6MroutesMapping . . . 1060
2.1233 show platform (fap | arad | jericho | jericho2) mroute ipv4 [...] . . . 1060
2.1234 show platform (fap | arad | jericho | jericho2) mroute ipv6 [...] . . . 1060
2.1235 show platform (fap | arad | jericho | jericho2) mvpn decap [...] . . . 1060
2.1235.1 MvpnDecapEntry . . . 1061
2.1236 show platform (fap | arad | jericho | jericho2) phy PHY_FROM_PORT dump [...] . . . 1061
2.1236.1 PhysicalInfo . . . 1061
2.1237 show platform (fap | arad | jericho | jericho2) phy PHY_FROM_PORT register mmd PHY_REG_START [...] . . . 1061
2.1237.1 PhysicalRegister . . . 1061
2.1238 show platform (fap | arad | jericho | jericho2) pmf [...] . . . 1062
2.1238.1 ProfileInstance . . . 1062
2.1239 show platform (fap | arad | jericho | jericho2) pmf dbinfo [...] . . . 1080

2.1239.1 TcamDbInfoTable . . . 1080
2.1240 show platform (fap | arad | jericho | jericho2) ppmc multicast [...] . . . 1081
2.1240.1 PpmcEntries . . . 1081
2.1241 show platform (fap | arad | jericho | jericho2) priority-flow-control buffer profile [...] . . . 1082
2.1241.1 FapConfig . . . 1082
2.1241.2 GlobalConfig . . . 1082
2.1242 show platform (fap | arad | jericho | jericho2) ptp hardware-sync [...] . . . 1083
2.1242.1 SlaveModel . . . 1083
2.1243 show platform (fap | arad | jericho | jericho2) qos tcam mapping [...] . . . 1084
2.1243.1 FapQosAclTcamMappingModel . . . 1084
2.1244 show platform (fap | arad | jericho | jericho2) qos tcam mapping bank BANK ENTRY [...] . . . 1086
2.1244.1 FapQosAclTcamEntryModel . . . 1086
2.1245 show platform (fap | arad | jericho | jericho2) qos tcam mapping policy-map PMAP [...] . . . 1086
2.1246 show platform (fap | arad | jericho | jericho2) sflow accel [...] . . . 1087
2.1246.1 PlatformSflowAccelResourceInfo . . . 1087
2.1247 show platform (fap | arad | jericho | jericho2) sflow accel counters [...] . . . 1087
2.1247.1 PlatformSflowAccelCountersInfo . . . 1087
2.1248 show platform (fap | arad | jericho | jericho2) subinterface mapping [...] . . . 1087
2.1248.1 SubInterfaceMap . . . 1088
2.1249 show platform (fap | arad | jericho | jericho2) subinterface mapping l2 [...] . . . 1088
2.1249.1 SubInterfaceL2Map . . . 1088
2.1250 show platform (fap | arad | jericho | jericho2) subinterface mapping l3 [...] . . . 1088
2.1250.1 SubInterfaceL3Map . . . 1089
2.1251 show platform (fap | arad | jericho | jericho2) tcam action-priority [...] . . . 1089
2.1251.1 TcamFapActionPriorityInfo . . . 1089
2.1252 show platform (fap | arad | jericho | jericho2) tcam summary [...] . . . 1089
2.1252.1 TcamFapInfo . . . 1089
2.1253 show platform (fap | arad | jericho | jericho2) vpls pseudowire decap [...] . . . 1090
2.1253.1 VplsPseudowireDetails . . . 1090
2.1254 show platform (fap | arad | jericho | jericho2) vrf selection inlif [...] . . . 1090
2.1254.1 VrfSelectionInLifBank . . . 1091
2.1255 show platform (fap | arad | jericho | jericho2) vrf selection tcam shadow [...] . . . 1092
2.1255.1 VrfSelectionExtendedV6TcamEntries . . . 1092
2.1255.2 VrfSelectionV4TcamEntries . . . 1092
2.1255.3 VrfSelectionV6TcamEntries . . . 1092
2.1256 show platform (fap | arad | jericho | jericho2) vxlan multicast route underlay [...] . . . 1093
2.1256.1 VxlanUnderlayMrouteEntry . . . 1093
2.1257 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric [...] . . . 1094
2.1257.1 FapFeSerdesInfo . . . 1094
2.1258 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric detail [...] . 1095
2.1259 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric rates mbps [...]1095
2.1260 show platform (fap | jericho2) mac security [...] . . . 1096
2.1260.1 MacSecTableFap . . . 1096
2.1261 show platform (fe | fe1600 | fe3600 | ramon) [...] . . . 1096
2.1261.1 FeInfo . . . 1096
2.1262 show platform (sand | arad) lag mode . . . 1097
2.1263 show platform agent-state verify (AGENT | all) . . . 1097
2.1263.1 PerAgentVerifyResultDict . . . 1097
2.1264 show platform barefoot acl . . . 1097
2.1264.1 ProgrammingStatusEntry . . . 1097
2.1265 show platform barefoot acl crlid . . . 1097
2.1265.1 CrlIdSummaryEntry . . . 1098
2.1266 show platform barefoot acl crlid active interface . . . 1098
2.1266.1 CrlIdChipActiveIntf . . . 1098
2.1267 show platform barefoot acl crlid active subinterface . . . 1098

2.1267.1 CrlIdChipActiveSubIntf . . . 1098
2.1268 show platform barefoot acl crlid active vlan . . . 1099
2.1268.1 CrlIdChipActiveVlan . . . 1099
2.1269 show platform barefoot acl crlid table racl . . . 1099
2.1269.1 CrlIdChipTableRacl . . . 1099
2.1270 show platform barefoot acl l4port . . . 1099
2.1270.1 L4PortChipSummary . . . 1099
2.1271 show platform barefoot acl prefix . . . 1100
2.1271.1 PrefixChipSummary . . . 1100
2.1272 show platform barefoot agent restarts . . . 1100
2.1272.1 BfnChipRestartsPerSourceModel . . . 1100
2.1273 show platform barefoot bfd-offload intervals . . . 1101
2.1274 show platform barefoot bfd-offload session DISCRIMINATOR | summary . . . 1101
2.1274.1 BfnBfdOffloadSession . . . 1101
2.1275 show platform barefoot counters nat interfaces [...] . . . 1101
2.1275.1 InterfacesCountersNatModel . . . 1101
2.1276 show platform barefoot counters table [...] . . . 1102
2.1276.1 PipeTableCounter . . . 1102
2.1277 show platform barefoot int drop codes . . . 1102
2.1277.1 DropReasonCodeEntry . . . 1102
2.1278 show platform barefoot l2 mac address-table . . . 1102
2.1278.1 VlanMacAddressTable . . . 1102
2.1279 show platform barefoot l2 mac address-table diff . . . 1103
2.1279.1 VlanMacAddress . . . 1103
2.1280 show platform barefoot l2 mac address-table pending . . . 1104
2.1280.1 PendingMacTableEntry . . . 1104
2.1281 show platform barefoot l2 stp blocked [...] . . . 1104
2.1281.1 PortsInVlan . . . 1104
2.1282 show platform barefoot l2 summary . . . 1104
2.1282.1 TableSummary . . . 1105
2.1283 show platform barefoot l3 (routes-v4 | routes-v6) [...] . . . 1105
2.1283.1 IpRoutes . . . 1105
2.1284 show platform barefoot l3 (routes-v4 | routes-v6) detail [...] . . . 1106
2.1284.1 IpRoutesDetail . . . 1106
2.1285 show platform barefoot l3 (vteps-v4 | vteps-v6) . . . 1106
2.1285.1 VtepEntry . . . 1107
2.1286 show platform barefoot l3 events . . . 1107
2.1286.1 IpVersionEvents . . . 1107
2.1287 show platform barefoot l3 interface . . . 1107
2.1287.1 InterfaceEntry . . . 1107
2.1288 show platform barefoot l3 mtu . . . 1108
2.1288.1 MtuEntry . . . 1108
2.1289 show platform barefoot l3 next-hop . . . 1108
2.1290 show platform barefoot l3 next-hop NEXTHOPID . . . 1108
2.1291 show platform barefoot l3 pipe allocation . . . 1108
2.1291.1 PipeResources . . . 1108
2.1292 show platform barefoot l3 route PREFIX [...] . . . 1108
2.1292.1 BackoffInfo . . . 1109
2.1292.2 RouteConfigInfo . . . 1109
2.1293 show platform barefoot l3 route last-hit (vrf (VRF | all)) [...] . . . 1109
2.1293.1 VrfRoutesLastHit . . . 1109
2.1294 show platform barefoot l3 route last-hit START_TIME END_TIME [...] . . . 1109
2.1295 show platform barefoot l3 route last-hit summary [...] . . . 1109
2.1295.1 NumRoutes . . . 1110
2.1296 show platform barefoot l3 scheduler [...] . . . 1110

2.1297 show platform barefoot l3 scheduler backlog TASKNAME . . . 1110
2.1297.1 RouteBacklogEntry . . . 1110
2.1298 show platform barefoot l3 summary . . . 1111
2.1298.1 PartitionUsage . . . 1111
2.1299 show platform barefoot l3 summary pipes [...] . . . 1111
2.1299.1 PipeUsageInfo . . . 1111
2.1300 show platform barefoot l3 task stats summary . . . 1111
2.1300.1 TaskStatsSummaryEntry . . . 1112
2.1301 show platform barefoot l3 vrf-mapping [...] . . . 1112
2.1301.1 VrfToPipe . . . 1112
2.1302 show platform barefoot lb errors . . . 1112
2.1303 show platform barefoot lb hash DIP SIP DPORT SPORT PROTO . . . 1112
2.1304 show platform barefoot lb nexthop . . . 1112
2.1304.1 LbNexthopEntry . . . 1112
2.1305 show platform barefoot lb nexthop-group NEXTHOP_GROUP hash HASH . . . 1113
2.1305.1 LbConnectionHistoryVersionInfo . . . 1113
2.1306 show platform barefoot lb nexthop-group NEXTHOP_GROUP_NAME . . . 1114
2.1306.1 L3Section . . . 1114
2.1306.2 LbSection . . . 1114
2.1307 show platform barefoot lb nexthop-group usage . . . 1114
2.1307.1 LbNhgUsageEntry . . . 1115
2.1308 show platform barefoot lb summary . . . 1115
2.1309 show platform barefoot load-balance profile . . . 1116
2.1309.1 BfnHashProfile . . . 1116
2.1310 show platform barefoot mirroring session . . . 1117
2.1310.1 BfnMirrorSession . . . 1117
2.1311 show platform barefoot port info . . . 1118
2.1311.1 PortInfoEntry . . . 1118
2.1312 show platform barefoot profile . . . 1118
2.1312.1 BfnProfileChipsPipeRoleModel . . . 1118
2.1313 show platform barefoot segment-security counters drop . . . 1118
2.1313.1 VrfHardwareDropCounters . . . 1118
2.1314 show platform barefoot sflow summary . . . 1119
2.1314.1 SflowConfigAndHwState . . . 1119
2.1315 show platform barefoot system . . . 1119
2.1315.1 BfnSystemEntry . . . 1119
2.1316 show platform barefoot traffic-manager queue status [...] . . . 1119
2.1316.1 QueuesInIntf . . . 1119
2.1317 show platform barefoot vlan membership DIRECTION . . . 1120
2.1318 show platform barefoot vlan table l2DomainLookup . . . 1120
2.1318.1 DeviceL2DomainLookupTable . . . 1120
2.1319 show platform barefoot vlan table vlanEncap . . . 1120
2.1319.1 DeviceVlanEncapTable . . . 1121
2.1320 show platform barefoot vlan table vlanProperties . . . 1121
2.1320.1 VlanPropertiesEntry . . . 1121
2.1321 show platform barefoot vxlan table intfOverlaySrcMacRewrite [...] . . . 1121
2.1321.1 VtiVniToMacAndHwState . . . 1121
2.1322 show platform barefoot vxlan table overlaySrcMacRewrite [...] . . . 1122
2.1322.1 VrfIpToMac . . . 1122
2.1323 show platform bios . . . 1123
2.1323.1 BiosVersion . . . 1123
2.1323.2 BiosVersions . . . 1123
2.1324 show platform bios history [...] . . . 1123
2.1324.1 BiosHistory . . . 1124
2.1325 show platform cortina INTF diag counters . . . 1124

2.1325.1 CountersModel . . . 1124
2.1326 show platform etba counters [...] . . . 1124
2.1326.1 HandlerCounter . . . 1124
2.1327 show platform fap ecmp hash [...] . . . 1125
2.1327.1 HashInfoFap . . . 1125
2.1328 show platform fap port-channel hash [...] . . . 1125
2.1329 show platform fap voq delete . . . 1125
2.1329.1 VoqDeleteEntry . . . 1125
2.1330 show platform ip igmp snooping groups [...] . . . 1125
2.1330.1 VlanStatus . . . 1125
2.1331 show platform module [...] . . . 1126
2.1331.1 PlatformModuleStatus . . . 1126
2.1332 show platform pkt . . . 1126
2.1333 show platform power controller . . . 1126
2.1333.1 PowerController . . . 1126
2.1334 show platform power isl . . . 1127
2.1334.1 IslController . . . 1127
2.1335 show platform ptp hardware-sync . . . 1127
2.1335.1 MasterModel . . . 1127
2.1336 show platform ptp hardware-sync master SCD_NAME . . . 1128
2.1337 show platform ptp hardware-sync slave SWITCH_NAME . . . 1128
2.1338 show platform sand capabilities . . . 1129
2.1339 show platform sand fabric monitor capacity . . . 1130
2.1339.1 FapFabricCapacityInfo . . . 1131
2.1340 show platform sand health . . . 1132
2.1340.1 EventLogModel . . . 1132
2.1340.2 FabricInterruptDataModel . . . 1132
2.1340.3 FapDataModel . . . 1132
2.1340.4 PeerLinkId . . . 1132
2.1341 show platform sand lag hardware-only . . . 1133
2.1342 show platform sand mcast replication ingress maximum . . . 1133
2.1343 show platform sand mdb profile [...] . . . 1133
2.1343.1 MdbFapProfile . . . 1133
2.1344 show platform sand scheduler interface INTF . . . 1134
2.1344.1 ClScheduler . . . 1134
2.1344.2 FqScheduler . . . 1134
2.1344.3 HrScheduler . . . 1134
2.1344.4 InterfaceSubmodel . . . 1135
2.1344.5 VoqIngressCore . . . 1135
2.1345 show platform sand subinterface anchor [...] . . . 1135
2.1346 show platform sand subinterface capabilities . . . 1135
2.1347 show platform sand tapagg [...] . . . 1136
2.1347.1 TapAggGroup . . . 1136
2.1347.2 TapAggNexthopGroup . . . 1136
2.1347.3 TapPort . . . 1137
2.1347.4 ToolPort . . . 1137
2.1348 show platform sand tapagg truncation-profile . . . 1137
2.1348.1 TruncationProfileFapModel . . . 1137
2.1349 show platform sand traffic-class to network-qos map . . . 1138
2.1350 show platform sand voq tail-drop . . . 1139
2.1350.1 IngressThresholds . . . 1139
2.1351 show platform sand voq tail-drop fadt . . . 1140
2.1352 show platform sand voq tail-drop fadt status . . . 1140
2.1353 show platform scd satellite debug [...] . . . 1140
2.1353.1 SatelliteScd . . . 1140

2.1354 show platform sfe counters [...] . . . 1140
2.1354.1 Counter . . . 1140
2.1355 show platform sfe flow tracking counters . . . 1142
2.1356 show platform sfe ip route [...] . . . 1143
2.1356.1 AsyncGate . . . 1143
2.1356.2 AsyncPrefix . . . 1143
2.1357 show platform sfe ip route diff . . . 1144
2.1357.1 Gates . . . 1144
2.1357.2 ModuleTable . . . 1144
2.1357.3 PModStatusTable . . . 1145
2.1357.4 Fec . . . 1145
2.1358 show platform sfe ip route multicast . . . 1145
2.1358.1 McastGates . . . 1145
2.1359 show platform sfe ipv6 route [...] . . . 1145
2.1360 show platform sfe l3 next-hops [...] . . . 1145
2.1360.1 AsyncFecInfo . . . 1146
2.1361 show platform sfe l3 next-hops diff . . . 1146
2.1361.1 FecInfo . . . 1147
2.1362 show platform sfe l3 peer-group . . . 1147
2.1363 show platform sfe l3 software summary . . . 1147
2.1364 show platform sfe l3 unicast summary . . . 1148
2.1365 show platform sfe licensing . . . 1148
2.1366 show platform sfe mirroring . . . 1149
2.1366.1 MirrorSession . . . 1149
2.1367 show platform sfe mroute AF [...] . . . 1150
2.1367.1 MfibMrouteGroup . . . 1150
2.1368 show platform sfe nat counters . . . 1151
2.1369 show platform sfe nat module pool-mgr . . . 1154
2.1370 show platform sfe nat pool [...] . . . 1156
2.1370.1 PoolRange . . . 1156
2.1371 show platform sfe qos . . . 1156
2.1372 show platform sfe route-cache ip hitbit (vrf VRF) . . . 1156
2.1372.1 HitbitEntry . . . 1157
2.1373 show platform sfe route-cache ip hitbit summary . . . 1157
2.1374 show platform sfe route-cache summary . . . 1157
2.1375 show platform sfe status . . . 1158
2.1375.1 NicInfo . . . 1158
2.1376 show platform smbus counters . . . 1158
2.1376.1 DeviceModel . . . 1158
2.1377 show platform smbus counters errors . . . 1159
2.1377.1 DeviceErrorModel . . . 1159
2.1378 show platform tfa pbr nexthops . . . 1159
2.1378.1 ArfaPbrNexthopModel . . . 1159
2.1379 show platform trident ((counters raw) | (counters raw (interface INTERFACE) | (all (interface INTERFACE)))) [...] . . . 1159
2.1379.1 PlatformCounterInterfaceRawModel . . . 1159
2.1380 show platform trident agent [...] . . . 1160
2.1380.1 TridentAgentModel . . . 1160
2.1381 show platform trident agent port [...] . . . 1160
2.1381.1 ChipPortsModel . . . 1160
2.1382 show platform trident agent role [...] . . . 1161
2.1382.1 RoleModel . . . 1161
2.1383 show platform trident ces connection [...] . . . 1161
2.1383.1 ConnectionsModel . . . 1161
2.1384 show platform trident counters [...] . . . 1162

2.1384.1 PlatformCountTypeCountersModel . . . 1162
2.1385 show platform trident counters fast-polling [...] . . . 1173
2.1385.1 FastPollInfoDetail . . . 1173
2.1386 show platform trident counters fast-polling interface [...] . . . 1174
2.1387 show platform trident counters interface (FABRIC | ETH | UETH) . . . 1174
2.1388 show platform trident counters polling . . . 1174
2.1389 show platform trident counters sampling . . . 1174
2.1389.1 SampledCounterChipModel . . . 1174
2.1390 show platform trident directflow [...] . . . 1176
2.1390.1 StrataSliceInfo . . . 1176
2.1391 show platform trident ecmp-lag (software | shadow) [...] . . . 1177
2.1391.1 EcmpLagStatus . . . 1177
2.1392 show platform trident egress event debug [...] . . . 1177
2.1392.1 EgrEventListModel . . . 1177
2.1393 show platform trident exact-match [...] . . . 1178
2.1393.1 StrataEmCliModelEntry . . . 1178
2.1394 show platform trident exact-match summary . . . 1179
2.1394.1 StrataEmCliSummaryModelEntry . . . 1179
2.1395 show platform trident fabric counters fec [...] . . . 1179
2.1395.1 PlatformSliceFecCounterModel . . . 1179
2.1396 show platform trident fabric counters queue [...] . . . 1179
2.1396.1 StrataFabricQueueCounterDropModel . . . 1179
2.1397 show platform trident fabric counters queue detail [...] . . . 1180
2.1397.1 StrataFabricInterfaceCounterModel . . . 1180
2.1398 show platform trident fabric counters rate . . . 1180
2.1399 show platform trident flexcounters l3intf in summary . . . 1180
2.1399.1 L3IntfInFlexCounterInfo . . . 1180
2.1400 show platform trident flexcounters l3intf in values . . . 1181
2.1400.1 FlexCounterValuesColl . . . 1181
2.1401 show platform trident flexcounters l3intf out summary . . . 1181
2.1402 show platform trident flexcounters l3intf out values . . . 1181
2.1403 show platform trident flexcounters multicast summary . . . 1181
2.1403.1 McastSources . . . 1181
2.1404 show platform trident flexcounters multicast values . . . 1182
2.1404.1 McastSources . . . 1182
2.1405 show platform trident flexcounters requests . . . 1184
2.1405.1 FlexRequest . . . 1184
2.1406 show platform trident flexcounters requests summary . . . 1185
2.1406.1 FlexRequestSummary . . . 1185
2.1407 show platform trident flexcounters summary . . . 1185
2.1407.1 PoolInfoList . . . 1185
2.1408 show platform trident flexcounters vni egress summary . . . 1186
2.1408.1 VniFlexCounterInfo . . . 1186
2.1409 show platform trident flexcounters vni egress values . . . 1186
2.1410 show platform trident flexcounters vni ingress summary . . . 1186
2.1411 show platform trident flexcounters vni ingress values . . . 1186
2.1412 show platform trident flexcounters vtep decap summary . . . 1186
2.1412.1 VtepDecapFlexCounterInfo . . . 1186
2.1413 show platform trident flexcounters vtep decap values . . . 1186
2.1414 show platform trident flexcounters vtep encap summary . . . 1187
2.1414.1 VtepFlexCounterInfo . . . 1187
2.1415 show platform trident flexcounters vtep encap values . . . 1187
2.1416 show platform trident flow tracker collector software . . . 1187
2.1416.1 Collector . . . 1187
2.1417 show platform trident flow tracker counters [...] . . . 1187

2.1417.1 StftOneLinecardCounter . . . 1187
2.1418 show platform trident flow tracker group software . . . 1189
2.1418.1 FlowGroup . . . 1189
2.1419 show platform trident flow tracker summary . . . 1189
2.1419.1 OneLinecardFlowTracker . . . 1190
2.1420 show platform trident forwarding-table partition . . . 1190
2.1421 show platform trident forwarding-table partition flexible . . . 1192
2.1422 show platform trident forwardingplane featureagents . . . 1193
2.1422.1 AgentModel . . . 1193
2.1423 show platform trident ingress event debug [...] . . . 1193
2.1423.1 IngEventListModel . . . 1193
2.1424 show platform trident interface map [...] . . . 1194
2.1424.1 ChipEntry . . . 1194
2.1425 show platform trident l2 hardware mac-address-table [...] . . . 1194
2.1425.1 MacAddressTable . . . 1194
2.1426 show platform trident l2 hardware mac-address-table diff [...] . . . 1195
2.1426.1 MacTableDiffEntry . . . 1195
2.1427 show platform trident l2 hardware mac-address-table missing [...] . . . 1195
2.1427.1 MacTableMissingEntry . . . 1195
2.1428 show platform trident l2 shadow counter [...] . . . 1196
2.1428.1 CounterTable . . . 1196
2.1429 show platform trident l2 shadow egr-tag-action-profile . . . 1196
2.1429.1 EgrVlanTagActionProfileEntry . . . 1196
2.1430 show platform trident l2 shadow egr-vfi . . . 1196
2.1430.1 EgrVfiEntry . . . 1197
2.1431 show platform trident l2 shadow egr-vlan-xlate [...] . . . 1198
2.1431.1 EgrVlanXlateHwTable . . . 1198
2.1432 show platform trident l2 shadow egr-vlan-xlate-2 [...] . . . 1200
2.1432.1 EgrVlanXlate2HwTable . . . 1200
2.1433 show platform trident l2 shadow ing-tag-action-profile . . . 1201
2.1433.1 IngVlanTagActionProfileEntry . . . 1201
2.1434 show platform trident l2 shadow source-vp . . . 1202
2.1434.1 VxlanSourceVpEntry . . . 1202
2.1435 show platform trident l2 shadow vfi . . . 1202
2.1435.1 VfiEntry . . . 1202
2.1436 show platform trident l2 shadow vlan-xlate [...] . . . 1204
2.1436.1 VlanXlateHwTable . . . 1204
2.1437 show platform trident l2 shadow vlan-xlate-2 [...] . . . 1205
2.1437.1 VlanXlate2HwTable . . . 1205
2.1438 show platform trident l2 software egr-vlan-xlate-2 . . . 1205
2.1438.1 L2SwEgrVlanXlate2Chip . . . 1205
2.1439 show platform trident l2 software ingress-vlan-range . . . 1206
2.1439.1 VlanRangeEntry . . . 1206
2.1440 show platform trident l2 software mac-address-table . . . 1206
2.1440.1 L2SwMacAddrTableEntry . . . 1206
2.1441 show platform trident l2 software mac-address-table count . . . 1207
2.1442 show platform trident l2 software mac-address-table remote count . . . 1207
2.1443 show platform trident l2 software multicast-group-table . . . 1208
2.1443.1 L2SwMcastGroupEntry . . . 1208
2.1443.2 L2SwMcastGroupTableCount . . . 1208
2.1444 show platform trident l2 software multicast-group-table count . . . 1208
2.1445 show platform trident l2 software vlan-xlate-2 . . . 1208
2.1445.1 L2SwVlanXlate2Chip . . . 1208
2.1446 show platform trident l2 status [...] . . . 1209
2.1446.1 L2StatusMacProgrammerStatus . . . 1209

2.1446.2 L2StatusMcastStatus . . . 1210
2.1447 show platform trident l3 cmdring . . . 1211
2.1447.1 CmdRing . . . 1211
2.1447.2 L3AgentStatus . . . 1215
2.1448 show platform trident l3 shadow (my-station | my-station-2) [...] . . . 1216
2.1448.1 L3MyStationEntry . . . 1216
2.1449 show platform trident l3 shadow counter [...] . . . 1216
2.1450 show platform trident l3 shadow dlb-ecmp-group-control [...] . . . 1216
2.1450.1 L3ShwDlbEcmpGroupControlEntry . . . 1216
2.1451 show platform trident l3 shadow dlb-ecmp-group-membership [...] . . . 1217
2.1451.1 L3ShwDlbEcmpGroupMembershipEntry . . . 1217
2.1452 show platform trident l3 shadow dlb-ecmp-group-port-to-member [...] . . . 1217
2.1452.1 L3ShwDlbEcmpGroupPortToMemberEntry . . . 1217
2.1453 show platform trident l3 shadow ecmp-dlb-control [...] . . . 1217
2.1453.1 L3ShwEcmpDlbControlEntry . . . 1217
2.1454 show platform trident l3 shadow interfaces [...] . . . 1217
2.1454.1 L3ShwIntfEntry . . . 1218
2.1455 show platform trident l3 shadow l3-iif-profile [...] . . . 1218
2.1455.1 L3ShwIifProfileEntry . . . 1218
2.1456 show platform trident l3 shadow l3-tunnel [...] . . . 1218
2.1456.1 L3ShwL3TunnelEntry . . . 1218
2.1457 show platform trident l3 shadow mroutes [...] . . . 1219
2.1457.1 L3ShwMroutesEntry . . . 1219
2.1458 show platform trident l3 shadow mtu [...] . . . 1219
2.1458.1 L3ShwMtuEntry . . . 1219
2.1459 show platform trident l3 shadow next-hops her . . . 1219
2.1459.1 L3ShwNextHopsHerEntry . . . 1220
2.1460 show platform trident l3 shadow next-hops multipath [...] . . . 1220
2.1460.1 L3NextHopsEcmp . . . 1220
2.1461 show platform trident l3 shadow next-hops multipath hfec [...] . . . 1220
2.1461.1 L3NextHopsEcmpHierarchicalList . . . 1220
2.1462 show platform trident l3 shadow next-hops simple [...] . . . 1220
2.1462.1 L3ShwNextHopsSimpleEntry . . . 1221
2.1463 show platform trident l3 shadow routes defip [...] . . . 1222
2.1463.1 L3ShwRoutesDefipEntry . . . 1222
2.1464 show platform trident l3 shadow routes host [...] . . . 1222
2.1464.1 L3ShwRoutesHostEntry . . . 1222
2.1465 show platform trident l3 shadow routes lpm [...] . . . 1224
2.1465.1 L3ShwRoutesLpmEntry . . . 1224
2.1466 show platform trident l3 shadow routes-v6 defip [...] . . . 1224
2.1467 show platform trident l3 shadow routes-v6 host [...] . . . 1224
2.1468 show platform trident l3 shadow routes-v6 lpm [...] . . . 1224
2.1469 show platform trident l3 shadow tunnel TYPE [...] . . . 1224
2.1469.1 L3ShwEgrIpTunnelEntry . . . 1225
2.1470 show platform trident l3 shadow unified-tcam . . . 1225
2.1470.1 UnifiedTcamEntry . . . 1225
2.1471 show platform trident l3 shadow unified-tcam map . . . 1225
2.1471.1 UnifiedTcamMapEntry . . . 1225
2.1472 show platform trident l3 software alpm-table [...] . . . 1226
2.1472.1 AlpmTableState . . . 1226
2.1472.2 L3AlpmRouteLpm . . . 1227
2.1472.3 L3AlpmRoutes . . . 1227
2.1472.4 L3AlpmTableBuckets . . . 1227
2.1472.5 L3AlpmThreeLevelTableStats . . . 1228
2.1472.6 L3AlpmTwoLevelTableStats . . . 1229

2.1473 show platform trident l3 software alpm-table routes-v6 [...] . . . 1231
2.1474 show platform trident l3 software ecmp . . . 1231
2.1474.1 simpleUcastEcmpBlocks . . . 1231
2.1475 show platform trident l3 software host-table [...] . . . 1232
2.1475.1 HostBankBucketInfo . . . 1232
2.1475.2 HostBankInfo . . . 1232
2.1476 show platform trident l3 software interfaces tunnel decap [...] . . . 1233
2.1476.1 L3ShwL3IntfTunnelEntry . . . 1233
2.1477 show platform trident l3 software interfaces tunnel encap [...] . . . 1233
2.1477.1 L3ShwEgrIpIntfTunnelEntry . . . 1233
2.1478 show platform trident l3 software l3-iif-profile . . . 1233
2.1478.1 IifTableRow . . . 1234
2.1479 show platform trident l3 software mgroups [...] . . . 1235
2.1479.1 MGroupModel . . . 1235
2.1480 show platform trident l3 software next-hops [...] . . . 1237
2.1480.1 L3SoftwareNextHopsHwHer . . . 1237
2.1480.2 L3SoftwareNextHopsMcast . . . 1238
2.1480.3 L3SoftwareNextHopsSdTag . . . 1238
2.1480.4 L3SoftwareNextHopsUcast . . . 1238
2.1481 show platform trident l3 software routes [...] . . . 1241
2.1481.1 StrataL3Vrfs . . . 1241
2.1482 show platform trident l3 software routes-v6 [...] . . . 1241
2.1483 show platform trident l3 software routing-vlan [...] . . . 1241
2.1483.1 PerVlanEntry . . . 1242
2.1484 show platform trident l3 software urpf exception-list [...] . . . 1242
2.1484.1 UnsupportedAclRule . . . 1242
2.1484.2 UrpfException . . . 1242
2.1485 show platform trident l3 summary . . . 1243
2.1485.1 McastModule . . . 1246
2.1486 show platform trident lag [...] . . . 1246
2.1486.1 LagDetailSingleEntry . . . 1246
2.1487 show platform trident lag detail [...] . . . 1246
2.1488 show platform trident lag nonucast-trunk-block-mask l3mc [...] . . . 1246
2.1488.1 NonUcastTrunkBlockMaskTbl . . . 1247
2.1489 show platform trident lag software . . . 1247
2.1490 show platform trident lanz coprocessor status . . . 1248
2.1490.1 CopStatusEntry . . . 1248
2.1491 show platform trident lanz hardware status . . . 1249
2.1491.1 HwStatusEntry . . . 1250
2.1492 show platform trident mac-address-table [...] . . . 1250
2.1493 show platform trident mac-address-table diff [...] . . . 1250
2.1494 show platform trident mac-address-table missing [...] . . . 1250
2.1495 show platform trident mac-traffic-policy hardware-id summary . . . 1250
2.1495.1 MatchRuleHardwareSummary . . . 1250
2.1496 show platform trident mac-traffic-policy rules status [...] . . . 1251
2.1496.1 MacTrafficPolicyRulesChipStatus . . . 1251
2.1497 show platform trident mirror index [...] . . . 1251
2.1497.1 IndexTable . . . 1251
2.1498 show platform trident mirror session [...] . . . 1251
2.1498.1 SessionInfo . . . 1251
2.1499 show platform trident mmu buffers status . . . 1252
2.1499.1 StrataMmuBufferModel . . . 1252
2.1500 show platform trident mmu queue cpu-profile [...] . . . 1252
2.1500.1 CpuQueueConfig . . . 1252
2.1501 show platform trident mmu queue headroom . . . 1253

2.1502 show platform trident mmu queue status [...] . . . 1253
2.1502.1 PlatformInterfaceMmuStatusModel . . . 1253
2.1503 show platform trident mmu queue status cpu [...] . . . 1254
2.1503.1 PlatformCpuStatusModel . . . 1254
2.1504 show platform trident mmu queue watermark [...] . . . 1255
2.1504.1 StrataInterfaceMmuIngressWatermarkModel . . . 1255
2.1504.2 StrataInterfaceMmuMulticastSharedWatermarkModel . . . 1255
2.1504.3 StrataInterfaceMmuUnicastSharedWatermarkModel . . . 1256
2.1504.4 StrataMmuIngressHeadroomPoolWatermarkModel . . . 1256
2.1505 show platform trident mmu queue watermark cpu [...] . . . 1256
2.1505.1 StrataCpuMmuQueueWatermarkModel . . . 1257
2.1506 show platform trident parity-event [...] . . . 1257
2.1506.1 ChipParityInformation . . . 1257
2.1507 show platform trident priority-flow-control watchdog snapshot [...] . . . 1258
2.1507.1 IntfSnapshotModel . . . 1258
2.1507.2 SnapshotSummaryModel . . . 1258
2.1508 show platform trident profile [...] . . . 1259
2.1508.1 ChipProfileModel . . . 1259
2.1508.2 TileScale . . . 1259
2.1509 show platform trident ptp hardware-sync [...] . . . 1260
2.1510 show platform trident security . . . 1260
2.1510.1 CommonStatusModel . . . 1260
2.1510.2 EgressStatusModel . . . 1260
2.1510.3 IngressStatusModel . . . 1260
2.1511 show platform trident security interface INTF . . . 1261
2.1511.1 L2Model . . . 1261
2.1511.2 L3Model . . . 1261
2.1511.3 PktDrvModel . . . 1261
2.1511.4 SecurityEngineModel . . . 1261
2.1511.5 SliceModel . . . 1261
2.1511.6 TcamModel . . . 1261
2.1512 show platform trident security interface INTF detail . . . 1262
2.1513 show platform trident switch-extender counters queue egress . . . 1262
2.1513.1 SwitchExtenderInterfaceModel . . . 1262
2.1514 show platform trident switch-extender counters queue ingress [...] . . . 1262
2.1514.1 SwitchExtenderInterfaceBackplaneModel . . . 1263
2.1515 show platform trident tcam cpu-bound [...] . . . 1264
2.1515.1 PoliciesModel . . . 1264
2.1516 show platform trident tcam profile [...] . . . 1267
2.1516.1 TcamChipInfo . . . 1267
2.1517 show platform trident tcam qos [...] . . . 1268
2.1518 show platform trident tcam status [...] . . . 1268
2.1518.1 AggregatorFeatureInfo . . . 1268
2.1519 show platform trident tcam summary [...] . . . 1269
2.1519.1 StrataTcamSummaryChip . . . 1269
2.1520 show platform trident vfi mc-index . . . 1269
2.1520.1 L3VfiMcIndexInfo . . . 1269
2.1521 show platform trident vfi virtual-ports . . . 1270
2.1521.1 L3VfiPhysicalInterfaceVp . . . 1270
2.1521.2 L3VfiSystemInterfaceVp . . . 1270
2.1522 show platform trident vfi vlan map . . . 1270
2.1523 show platform trident vlan membership egr-vp . . . 1271
2.1523.1 VpVlanMembershipEntry . . . 1271
2.1524 show platform trident vlan membership egr-vp unprogrammed . . . 1271
2.1525 show platform trident vlan membership ing-vp . . . 1271

2.1526 show platform trident vlan membership ing-vp unprogrammed . . . 1271
2.1527 show platform trident vlan profile . . . 1271
2.1527.1 VlanProfileEntry . . . 1271
2.1528 show platform trident vlan setting . . . 1272
2.1528.1 VlanSettingEntry . . . 1272
2.1529 show platform trident vlan status . . . 1272
2.1529.1 VlanStatusEntry . . . 1272
2.1530 show platform trident vlan stg . . . 1272
2.1530.1 VlanStgEntry . . . 1273
2.1531 show platform trident vxlan local-vtep . . . 1273
2.1531.1 VxlanLocalVtepV4V6InterfaceModel . . . 1273
2.1532 show platform trident vxlan multicast route underlay [...] . . . 1273
2.1532.1 MrouteKeyToEMEntryModel . . . 1274
2.1533 show platform trident vxlan multicast tunnel [...] . . . 1274
2.1533.1 McastTunnelEntryModel . . . 1274
2.1534 show platform trident vxlan multihoming groups non-peering . . . 1274
2.1534.1 FilteredGroupRow . . . 1274
2.1535 show platform trident vxlan multihoming groups peering . . . 1274
2.1535.1 PeeringGroupRow . . . 1275
2.1536 show platform trident vxlan next-hop . . . 1275
2.1536.1 HerVtepNextHopsModel . . . 1275
2.1536.2 VfiNextHopsModel . . . 1275
2.1537 show platform trident vxlan port-to-next-hop . . . 1275
2.1537.1 VxlanPortToNhTable . . . 1276
2.1538 show platform trident vxlan port-to-next-hop overflow . . . 1276
2.1539 show platform trident vxlan routing-info . . . 1276
2.1539.1 VlanToSvpEntry . . . 1276
2.1540 show platform trident vxlan vlan [...] . . . 1277
2.1540.1 VxlanVlanEntry . . . 1277
2.1541 show platform trident vxlan vlan summary . . . 1277
2.1542 show platform trident vxlan vp-usage . . . 1277
2.1543 show platform trident vxlan vtep [...] . . . 1277
2.1543.1 VxlanRemoteVtepInfoModel . . . 1278
2.1544 show poe [...] . . . 1279
2.1544.1 PoePort . . . 1279
2.1545 show poe neighbor [...] . . . 1280
2.1545.1 PoePortNbrInfo . . . 1280
2.1546 show policing [...] . . . 1281
2.1546.1 PolicerInstanceModel . . . 1281
2.1546.2 PolicerPktSizeAdjProfileModel . . . 1281
2.1546.3 ProfileModel . . . 1281
2.1547 show policing interfaces [...] . . . 1283
2.1547.1 InterfacePolicerModel . . . 1283
2.1548 show policing interfaces counters [...] . . . 1283
2.1548.1 InterfacePolicingCounterModel . . . 1283
2.1549 show policy-map (copp | (interface control-plane)) PMAP . . . 1284
2.1550 show policy-map (copp | (interface control-plane)) copp-system-policy . . . 1285
2.1550.1 PolicyMapModel . . . 1285
2.1551 show policy-map PMAP class CMAP [...] . . . 1287
2.1552 show policy-map PMAP counters [...] . . . 1287
2.1553 show policy-map [...] . . . 1288
2.1554 show policy-map counters [...] . . . 1288
2.1555 show policy-map interface (INTF | SVI_INTF | VTI_INTF) [...] . . . 1288
2.1556 show policy-map interface INTF type control-plane . . . 1288
2.1557 show policy-map type (copp | control-plane) (copp-system-policy | PMAP) class CMAP . . . 1288

2.1558 show policy-map type (copp | control-plane) [...] . . . 1288
2.1559 show policy-map type (copp | control-plane) counters [...] . . . 1288
2.1560 show policy-map type pbr [...] . . . 1290
2.1560.1 PolicyMapModel . . . 1290
2.1561 show policy-map type tapagg [...] . . . 1300
2.1562 show port numbering . . . 1300
2.1563 show port-channel [...] . . . 1301
2.1564 show port-channel dense . . . 1302
2.1564.1 PortChannelSummary . . . 1302
2.1565 show port-channel limits . . . 1303
2.1565.1 LagGroupLimits . . . 1303
2.1566 show port-channel load-balance (jericho | jericho2) fields [...] . . . 1303
2.1567 show port-channel load-balance [...] . . . 1303
2.1567.1 PortChannelTraffic . . . 1303
2.1568 show port-channel load-balance distribution . . . 1304
2.1569 show port-channel load-balance trident fields . . . 1305
2.1569.1 LagHashFieldsCommon . . . 1305
2.1569.2 LagHashFieldsGre . . . 1305
2.1569.3 LagHashFieldsGtp . . . 1305
2.1569.4 LagHashFieldsIp . . . 1305
2.1569.5 LagHashFieldsIpv6 . . . 1306
2.1569.6 LagHashFieldsMac . . . 1306
2.1569.7 LagHashFieldsMim . . . 1306
2.1569.8 LagHashFieldsMpls . . . 1306
2.1570 show port-channel load-balance trident udf . . . 1307
2.1570.1 StrataLagUdfHashLoadBalanceModel . . . 1307
2.1571 show port-channel summary . . . 1307
2.1572 show port-channel traffic [...] . . . 1307
2.1573 show port-security . . . 1307
2.1573.1 PortStatistic . . . 1307
2.1574 show port-security address . . . 1308
2.1574.1 Address . . . 1308
2.1575 show port-security interface [...] . . . 1309
2.1575.1 Interface . . . 1309
2.1576 show port-security mac-address . . . 1310
2.1577 show port-security vlan [...] . . . 1310
2.1577.1 VlanDict . . . 1310
2.1578 show priority-flow-control [...] . . . 1310
2.1579 show priority-flow-control counters [...] . . . 1310
2.1580 show priority-flow-control counters detail [...] . . . 1310
2.1581 show priority-flow-control counters history [...] . . . 1311
2.1581.1 PfcIntfHistoryCountersModel . . . 1311
2.1582 show priority-flow-control counters watchdog [...] . . . 1311
2.1583 show priority-flow-control counters watchdog drop [...] . . . 1311
2.1584 show priority-flow-control interfaces Fabric status . . . 1311
2.1585 show priority-flow-control status [...] . . . 1311
2.1586 show privilege . . . 1312
2.1587 show processes . . . 1312
2.1587.1 ProcessInfo . . . 1312
2.1587.2 SystemTimeInfo . . . 1312
2.1588 show processes top once [...] . . . 1313
2.1588.1 CpuInfo . . . 1313
2.1588.2 ProcessActivityInfo . . . 1313
2.1588.3 SystemMemSum . . . 1314
2.1588.4 ThreadsInfo . . . 1314

2.1589 show ptp . . . 1314
2.1590 show ptp ((local-clock time-properties) | time-property) . . . 1314
2.1590.1 PtpTimeProperty . . . 1314
2.1591 show ptp (clock | local-clock) . . . 1316
2.1591.1 PtpOrdinaryOrBoundaryClock . . . 1316
2.1592 show ptp (masters | parent) . . . 1317
2.1592.1 PtpParent . . . 1317
2.1593 show ptp foreign-master-record . . . 1317
2.1593.1 PtpIntfVlanForeignMasterRecords . . . 1318
2.1594 show ptp hardware-sync . . . 1318
2.1595 show ptp hold-ptp-time . . . 1318
2.1596 show ptp interface INTF vlan VLAN_ID . . . 1320
2.1596.1 PtpIntfVlans . . . 1320
2.1597 show ptp interface [...] . . . 1323
2.1598 show ptp interface counters [...] . . . 1324
2.1598.1 PtpIntfVlansCounters . . . 1324
2.1599 show ptp interface counters drop [...] . . . 1324
2.1599.1 PtpDebugIntfCounters . . . 1324
2.1600 show ptp interface vlans [...] . . . 1326
2.1600.1 PtpVlansExtendedWrapper . . . 1326
2.1601 show ptp ip access-list . . . 1326
2.1602 show ptp management-plane . . . 1326
2.1603 show ptp monitor . . . 1327
2.1603.1 PtpMonitorDataEntry . . . 1327
2.1604 show ptp mpass . . . 1327
2.1605 show ptp mpass interface [...] . . . 1328
2.1606 show ptp mpass interface counters [...] . . . 1328
2.1607 show ptp source . . . 1328
2.1608 show ptp source ip . . . 1328
2.1609 show ptp unicast-negotiation candidate-grantor . . . 1328
2.1609.1 UcastNegCandidateGrantorEntry . . . 1328
2.1610 show ptp unicast-negotiation granted . . . 1328
2.1610.1 UcastNegMessages . . . 1329
2.1611 show ptp unicast-negotiation profile [...] . . . 1329
2.1611.1 UcastNegProfile . . . 1329
2.1612 show ptp unicast-negotiation remote-grantee . . . 1330
2.1612.1 UcastNegRemoteGranteeEntry . . . 1330
2.1613 show ptp unicast-negotiation requested . . . 1330
2.1614 show qos ecn [...] . . . 1330
2.1614.1 EcnDelayParametersModel . . . 1330
2.1614.2 EcnParametersModel . . . 1330
2.1615 show qos ecn counters . . . 1331
2.1616 show qos ecn delay [...] . . . 1331
2.1617 show qos interfaces [...] . . . 1332
2.1617.1 IntfQosAllModel . . . 1332
2.1618 show qos interfaces drop-thresholds [...] . . . 1336
2.1618.1 IntfDropThresholdsModel . . . 1336
2.1619 show qos interfaces dscp mpls encapsulation disabled [...] . . . 1337
2.1620 show qos interfaces ecn [...] . . . 1337
2.1620.1 IntfEcnModel . . . 1337
2.1621 show qos interfaces ecn counters queue [...] . . . 1337
2.1621.1 EcnQueueCounterModel . . . 1338
2.1622 show qos interfaces fabric [...] . . . 1338
2.1623 show qos interfaces latency maximum [...] . . . 1338
2.1623.1 IntfLatencyThresholdModel . . . 1338

2.1624 show qos interfaces random-detect drop [...] . . . 1338
2.1624.1 IntfWredModel . . . 1338
2.1625 show qos interfaces random-detect ecn [...] . . . 1339
2.1626 show qos interfaces trust [...] . . . 1339
2.1626.1 IntfTrustModeModel . . . 1339
2.1627 show qos interfaces wred counters queue [...] . . . 1340
2.1627.1 WredQueueCounterModel . . . 1340
2.1628 show qos map cos to traffic-class [...] . . . 1340
2.1628.1 CosToTcProfileModel . . . 1340
2.1629 show qos map dscp to traffic-class [...] . . . 1340
2.1629.1 DscpToTcMapModel . . . 1340
2.1630 show qos map exp to traffic-class [...] . . . 1341
2.1630.1 ExpToTcProfileModel . . . 1341
2.1631 show qos map traffic-class to cos [...] . . . 1341
2.1631.1 TcToCosMapModel . . . 1341
2.1632 show qos map traffic-class to dscp [...] . . . 1341
2.1632.1 DscpRewriteMapModel . . . 1341
2.1633 show qos maps [...] . . . 1342
2.1634 show qos profile [...] . . . 1344
2.1634.1 QosProfileModel . . . 1344
2.1635 show qos profile summary [...] . . . 1346
2.1635.1 QosProfileConfiguredIntfs . . . 1346
2.1636 show qos random-detect ecn [...] . . . 1347
2.1637 show qos scheduling group [...] . . . 1347
2.1637.1 IntfSchedulingGroupWithMembersModel . . . 1347
2.1638 show qos scheduling hierarchy [...] . . . 1347
2.1638.1 QosSchedulingHierarchyModel . . . 1347
2.1639 show queue-monitor length (all | global-buffer | statistics) . . . 1348
2.1639.1 CongestionRecordModel . . . 1348
2.1639.2 BarefootCongestionRecord . . . 1348
2.1639.3 BarefootStatisticsRecord . . . 1348
2.1639.4 MakoCongestionRecord . . . 1349
2.1639.5 MakoStatisticsRecord . . . 1349
2.1639.6 SandCongestionRecord . . . 1349
2.1639.7 SandLatencyRecord . . . 1349
2.1639.8 StrataCommonRecord . . . 1350
2.1639.9 StrataCongestionRecord . . . 1350
2.1639.10 StrataDropsRecord . . . 1351
2.1639.11 StrataLatencyRecord . . . 1351
2.1639.12 StrataStatisticsRecord . . . 1352
2.1639.13 XpCongestionRecord . . . 1352
2.1639.14 XpStatisticsRecord . . . 1352
2.1640 show queue-monitor length [...] . . . 1353
2.1641 show queue-monitor length csv [...] . . . 1353
2.1642 show queue-monitor length status . . . 1354
2.1642.1 InterfaceStatusModel . . . 1355
2.1642.2 MakoQueueMonitorLengthStatusModel . . . 1355
2.1643 show queue-monitor streaming ((ip access-list) | (ipv6 access-list)) [...] . . . 1355
2.1644 show queue-monitor streaming clients . . . 1355
2.1645 show radius . . . 1356
2.1645.1 RadiusStats . . . 1356
2.1645.2 ServerGroup . . . 1357
2.1646 show recirc-channel [...] . . . 1357
2.1647 show recirc-channel limits . . . 1357
2.1648 show recirc-channel traffic [...] . . . 1357

2.1649 show redundancy file-replication . . . 1358
2.1650 show redundancy states . . . 1359
2.1651 show redundancy status . . . 1359
2.1652 show redundancy switchover sso . . . 1359
2.1652.1 SwitchoverLog . . . 1360
2.1653 show redundancy time-sync . . . 1360
2.1654 show reload . . . 1360
2.1655 show reload cause [...] . . . 1360
2.1655.1 ReloadCause . . . 1360
2.1656 show reload cause history . . . 1360
2.1656.1 ReloadCauseList . . . 1361
2.1657 show reload console logs [...] . . . 1361
2.1658 show reload console status . . . 1361
2.1658.1 ConsoleLogFile . . . 1361
2.1659 show reload fast-boot . . . 1361
2.1659.1 ReloadHitlessBlockingReason . . . 1362
2.1659.2 ReloadHitlessWarningReason . . . 1363
2.1660 show reload hitless . . . 1363
2.1661 show rib loop (ip | ipv6) route [...] . . . 1364
2.1661.1 LoopedRoute . . . 1364
2.1662 show rib multicast route (ip | ipv6) [...] . . . 1364
2.1663 show rib next-hop (ip | ipv6) [...] . . . 1364
2.1663.1 RibNextHop . . . 1364
2.1664 show rib next-hop (ip | ipv6) dependency . . . 1364
2.1664.1 RecursiveNhDependencyByProtocol . . . 1365
2.1665 show rib next-hop resolution route (ipv4 | ipv6) [...] . . . 1365
2.1665.1 ResolutionRoute . . . 1365
2.1666 show rib ready [...] . . . 1367
2.1666.1 RibReadyModel . . . 1367
2.1667 show rib route (ip | ipv6) [...] . . . 1368
2.1668 show rib route (ip | ipv6) internal [...] . . . 1368
2.1669 show rib route (ip | ipv6) tag TAG [...] . . . 1369
2.1670 show rib route summary [...] . . . 1369
2.1670.1 RibRouteSummaryForVrf . . . 1369
2.1671 show rib route summary brief [...] . . . 1369
2.1672 show rib system-tunneling-ldp-rib route ipv4 [...] . . . 1369
2.1673 show role [...] . . . 1369
2.1673.1 Rules . . . 1369
2.1674 show route (ipv4 | ipv6) counters [...] . . . 1370
2.1674.1 RouteCounters . . . 1370
2.1675 show route cache (vrf (VRF | all)) [...] . . . 1370
2.1675.1 Prefix . . . 1370
2.1676 show route cache (vrf (VRF | all)) ipv6 [...] . . . 1371
2.1677 show route cache summary [...] . . . 1371
2.1677.1 RouteCacheSummary . . . 1371
2.1678 show route cache threshold . . . 1371
2.1678.1 ThresholdParams . . . 1371
2.1679 show route-map [...] . . . 1372
2.1680 show route-map config-sanity . . . 1372
2.1681 show route-map expanded [...] . . . 1372
2.1681.1 ExpandedRouteMap . . . 1372
2.1682 show router igp topology [...] . . . 1373
2.1682.1 VrfModel . . . 1373
2.1683 show router rcf code [...] . . . 1374
2.1684 show router rcf errors [...] . . . 1374

2.1685 show router rcf function location [...] . . . 1374
2.1685.1 FunctionLocationModel . . . 1374
2.1686 show router rcf pending . . . 1375
2.1686.1 RcfPendingCodeUnitModel . . . 1375
2.1687 show running-config [...] . . . 1375
2.1688 show running-config all [...] . . . 1375
2.1689 show running-config digest [...] . . . 1375
2.1690 show schedule NAME | summary . . . 1375
2.1690.1 ScheduleJobModels . . . 1375
2.1691 show security pki certificate rotation ROTATION_ID . . . 1376
2.1691.1 CertificateSigningRequest . . . 1376
2.1692 show security pki certificate rotation [...] . . . 1376
2.1693 show segment-security [...] . . . 1377
2.1693.1 Vrf . . . 1377
2.1694 show segment-security application [...] . . . 1377
2.1694.1 Service . . . 1378
2.1695 show segment-security counters [...] . . . 1378
2.1695.1 DestSegment . . . 1378
2.1695.2 PolicyCounter . . . 1379
2.1696 show segment-security errors [...] . . . 1379
2.1696.1 Vrf . . . 1379
2.1697 show segment-security hardware counters [...] . . . 1379
2.1697.1 VrfHardwareCounters . . . 1379
2.1698 show segment-security hardware detail [...] . . . 1380
2.1698.1 VrfHardwareDetail . . . 1380
2.1699 show segment-security hardware routes [...] . . . 1380
2.1699.1 VrfHardwareRoutes . . . 1381
2.1700 show segment-security hardware summary [...] . . . 1381
2.1700.1 VrfHardwareSummary . . . 1381
2.1701 show segment-security policy [...] . . . 1381
2.1701.1 Policy . . . 1381
2.1702 show segment-security status [...] . . . 1382
2.1702.1 Vrf . . . 1382
2.1703 show service mss status . . . 1382
2.1703.1 MssPolicyEnforcementModel . . . 1382
2.1704 show service pcs controller group [...] . . . 1383
2.1704.1 ControllerNsGroup . . . 1383
2.1705 show service pcs controller host IP_GEN_ADDR . . . 1383
2.1706 show service pcs controller policy [...] . . . 1383
2.1706.1 ControllerPolicy . . . 1383
2.1707 show service pcs policy [...] . . . 1384
2.1707.1 PolicySectionStatus . . . 1384
2.1708 show service pcs policy counters [...] . . . 1384
2.1708.1 PolicyCounters . . . 1384
2.1709 show service pcs status . . . 1385
2.1709.1 PcsApiState . . . 1385
2.1710 show service pcs tag [...] . . . 1385
2.1710.1 TagInfo . . . 1385
2.1711 show service vxlan address-table advertised [...] . . . 1386
2.1711.1 VxlanVniStatus . . . 1386
2.1712 show service vxlan address-table received [...] . . . 1386
2.1713 show service vxlan arp advertised [...] . . . 1387
2.1713.1 VxlanArpTable . . . 1387
2.1714 show service vxlan arp received [...] . . . 1387
2.1714.1 VxlanVtep . . . 1387

2.1715 show service vxlan nd advertised [...] . . . 1387
2.1716 show service vxlan nd received [...] . . . 1387
2.1717 show service vxlan status . . . 1387
2.1717.1 VxlanControllerCapability . . . 1388
2.1718 show service vxlan vni advertised [...] . . . 1388
2.1718.1 VniList . . . 1388
2.1719 show service vxlan vni received [...] . . . 1388
2.1719.1 SwitchList . . . 1389
2.1720 show service-insertion connection [...] . . . 1389
2.1720.1 SfeSiConnStatusModel . . . 1389
2.1721 show session-config [...] . . . 1389
2.1722 show sflow [...] . . . 1390
2.1722.1 BgpExport . . . 1391
2.1722.2 Ipv4Destination . . . 1391
2.1722.3 Ipv4Source . . . 1391
2.1722.4 Ipv6Destination . . . 1392
2.1722.5 Ipv6Source . . . 1392
2.1722.6 SendingDatagram . . . 1392
2.1722.7 Details . . . 1392
2.1723 show sflow hardware accelerators . . . 1393
2.1723.1 SflowAcceleratorInfo . . . 1393
2.1724 show sflow hardware counters . . . 1393
2.1724.1 HardwareSflowCountersInfo . . . 1393
2.1725 show sflow hardware mapping . . . 1394
2.1725.1 SflowAcceleratorMappingInfo . . . 1394
2.1726 show sflow hardware mapping lag . . . 1394
2.1726.1 SflowAcceleratorMappingLagInfo . . . 1394
2.1727 show sflow hardware status . . . 1394
2.1727.1 Vrf . . . 1394
2.1727.2 SflowAccelModuleStatus . . . 1395
2.1728 show sflow interfaces [...] . . . 1395
2.1728.1 LagMemberSflowTypeStatus . . . 1395
2.1729 show sflow interfaces counters . . . 1396
2.1729.1 IntfSampleCounterMap . . . 1396
2.1730 show snmp . . . 1397
2.1730.1 Contact . . . 1397
2.1730.2 SnmpChassis . . . 1397
2.1730.3 SnmpCounters . . . 1397
2.1730.4 SnmpInterface . . . 1398
2.1730.5 SnmpLocation . . . 1398
2.1730.6 SnmpLogging . . . 1398
2.1730.7 SnmpMiscConfig . . . 1398
2.1730.8 SnmpVrfs . . . 1398
2.1731 show snmp ((ipv4 access-list) | (ipv6 access-list)) [...] . . . 1398
2.1732 show snmp (notification | trap) . . . 1398
2.1732.1 SnmpNotification . . . 1399
2.1733 show snmp chassis . . . 1399
2.1734 show snmp community . . . 1399
2.1734.1 SnmpCommunityDetailModel . . . 1399
2.1735 show snmp contact . . . 1399
2.1736 show snmp debug counters table (TABLE | all) [...] . . . 1399
2.1736.1 SnmpTableIteratorCounterModel . . . 1400
2.1737 show snmp debug identifier hashed . . . 1400
2.1738 show snmp engineID . . . 1400
2.1738.1 SnmpRemoteEngineIdHostToPortModel . . . 1400

2.1739 show snmp group [...] . . . 1400
2.1739.1 SnmpGroupVerToDetailModel . . . 1401
2.1740 show snmp host . . . 1401
2.1740.1 HostModel . . . 1401
2.1741 show snmp local-interface . . . 1402
2.1742 show snmp location . . . 1402
2.1743 show snmp mib . . . 1402
2.1743.1 SnmpOidsModel . . . 1402
2.1744 show snmp mib bridgemib portid [...] . . . 1402
2.1745 show snmp mib ifmib ifindex [...] . . . 1402
2.1746 show snmp mib qosmib counter-interval . . . 1402
2.1747 show snmp mib translate OID . . . 1402
2.1748 show snmp net-snmp buffer . . . 1403
2.1749 show snmp notification host . . . 1403
2.1750 show snmp scheduler [...] . . . 1403
2.1751 show snmp source-interface . . . 1403
2.1752 show snmp user [...] . . . 1403
2.1752.1 UserCollectionModel . . . 1403
2.1753 show snmp v2-mib chassis . . . 1404
2.1754 show snmp v2-mib contact . . . 1404
2.1755 show snmp v2-mib location . . . 1404
2.1756 show snmp view [...] . . . 1404
2.1756.1 SnmpView . . . 1404
2.1757 show spanning-tree [...] . . . 1405
2.1757.1 SpanningTreeInstance . . . 1405
2.1758 show spanning-tree blockedports . . . 1408
2.1758.1 SpanningTreeBlockedPortsList . . . 1408
2.1759 show spanning-tree bridge [...] . . . 1409
2.1759.1 SpanningTreeBridgesDetail . . . 1409
2.1760 show spanning-tree bridge assurance [...] . . . 1410
2.1760.1 SpanningTreeBridgeAssuranceInterfaces . . . 1410
2.1761 show spanning-tree counters . . . 1410
2.1762 show spanning-tree instance [...] . . . 1410
2.1763 show spanning-tree interface ETHINTF [...] . . . 1410
2.1764 show spanning-tree mst [...] . . . 1410
2.1764.1 SpanningTreeMstInstance . . . 1411
2.1765 show spanning-tree mst configuration [...] . . . 1411
2.1766 show spanning-tree mst interface ETHINTF [...] . . . 1411
2.1766.1 SpanningTreeMstInterface . . . 1411
2.1767 show spanning-tree mst test information . . . 1413
2.1767.1 BridgeStpiInfo . . . 1413
2.1768 show spanning-tree root [...] . . . 1414
2.1768.1 SpanningTreeRoot . . . 1414
2.1769 show spanning-tree topology status [...] . . . 1414
2.1769.1 Topology . . . 1414
2.1770 show spanning-tree transmit active [...] . . . 1414
2.1771 show spanning-tree vlan [...] . . . 1415
2.1771.1 SpanningTreeVlanInstance . . . 1415
2.1772 show storm-control [...] . . . 1415
2.1772.1 IntfStormControlStatus . . . 1415
2.1773 show stun client counters [...] . . . 1416
2.1773.1 StunServerProfileCountersModel . . . 1416
2.1774 show stun client translations [...] . . . 1416
2.1774.1 StunClientBindingModel . . . 1416
2.1775 show stun server bindings [...] . . . 1417

2.1775.1 StunServerBindingModel . . . 1417
2.1776 show stun server status . . . 1417
2.1777 show switch forwarding-mode . . . 1417
2.1777.1 SupportedMode . . . 1418
2.1778 show switch scheduler . . . 1418
2.1779 show sync-e esmc counters . . . 1419
2.1779.1 SynceEsmcCountersInfo . . . 1419
2.1780 show sync-e esmc detail [...] . . . 1421
2.1780.1 SynceEsmcDetailInfo . . . 1421
2.1781 show sync-e selection . . . 1422
2.1781.1 SynceSelectionStatus . . . 1423
2.1782 show system bmc status . . . 1423
2.1783 show system coredump . . . 1423
2.1784 show system environment cooling [...] . . . 1423
2.1785 show system environment power . . . 1424
2.1786 show system environment power budget . . . 1424
2.1786.1 PowerConsumer . . . 1424
2.1787 show system environment power current . . . 1424
2.1788 show system environment power detail . . . 1424
2.1789 show system environment power histogram . . . 1424
2.1790 show system environment power voltage . . . 1424
2.1791 show system environment temperature [...] . . . 1425
2.1792 show system environment temperature transceiver [...] . . . 1425
2.1793 show system health storage . . . 1425
2.1793.1 StorageHealth . . . 1425
2.1794 show system storage health . . . 1425
2.1795 show system storage health smart . . . 1425
2.1795.1 DeviceSmartAttrs . . . 1426
2.1796 show system storage health smart detail . . . 1426
2.1796.1 DeviceSmartAttrDetails . . . 1426
2.1797 show system storage statistics . . . 1426
2.1797.1 IOStat . . . 1426
2.1798 show system storage statistics last day . . . 1427
2.1799 show system storage statistics rates . . . 1427
2.1800 show system systemd [...] . . . 1427
2.1800.1 SystemdUnitStatus . . . 1427
2.1801 show system systemd detail . . . 1428
2.1801.1 SystemdMountpointsDetail . . . 1428
2.1801.2 SystemdServicesDetail . . . 1428
2.1801.3 SystemdSocketsDetail . . . 1429
2.1802 show system systemd mountpoints detail . . . 1430
2.1803 show system systemd services detail . . . 1430
2.1804 show system systemd sockets detail . . . 1430
2.1805 show tacacs . . . 1431
2.1805.1 ServerGroup . . . 1431
2.1805.2 TacacsStats . . . 1431
2.1806 show tap aggregation counters default-forwarding . . . 1432
2.1806.1 CounterValue . . . 1432
2.1807 show tap aggregation counters tunnel termination gre . . . 1432
2.1808 show tap aggregation groups [...] . . . 1432
2.1808.1 TapAggGroup . . . 1432
2.1809 show tap aggregation groups map TAPPORT [...] . . . 1433
2.1809.1 MapPropertyList . . . 1433
2.1810 show tap aggregation groups map TOOLPORT [...] . . . 1433
2.1811 show tcpdump . . . 1433

2.1811.1 SessionModel . . . 1433
2.1812 show terminal . . . 1434
2.1813 show trace AGENT . . . 1435
2.1813.1 StateAndLevel . . . 1435
2.1814 show traffic-engineering cbf fec [...] . . . 1435
2.1814.1 CbfFecModel . . . 1435
2.1815 show traffic-engineering cspf path IP_ADDR links . . . 1437
2.1815.1 CspfPathLinksVrfModel . . . 1437
2.1816 show traffic-engineering cspf path PATH_ID [...] . . . 1440
2.1816.1 CspfPathByIdVrfModel . . . 1440
2.1817 show traffic-engineering cspf path PATH_ID links . . . 1441
2.1817.1 CspfPathByIdLinksVrfModel . . . 1441
2.1818 show traffic-engineering cspf path [...] . . . 1443
2.1818.1 CspfPathVrfModel . . . 1443
2.1819 show traffic-engineering database . . . 1444
2.1819.1 TeVrfDbModel . . . 1444
2.1820 show traffic-engineering database duplicate interface-address . . . 1445
2.1820.1 InterfaceIpInfoModel . . . 1445
2.1821 show traffic-engineering database router-id IP . . . 1445
2.1822 show traffic-engineering gribi ipv4-aft [...] . . . 1446
2.1822.1 GribiIpAft . . . 1446
2.1823 show traffic-engineering gribi ipv4-aft unacknowledged [...] . . . 1446
2.1823.1 GribiPendingIpAft . . . 1446
2.1824 show traffic-engineering gribi ipv6-aft [...] . . . 1447
2.1825 show traffic-engineering gribi ipv6-aft unacknowledged [...] . . . 1447
2.1826 show traffic-engineering gribi mpls-aft [...] . . . 1447
2.1826.1 GribiMplsAftType . . . 1447
2.1827 show traffic-engineering gribi nexthop-aft [...] . . . 1448
2.1827.1 GribiNhAftType . . . 1448
2.1828 show traffic-engineering gribi nexthop-aft unacknowledged [...] . . . 1448
2.1828.1 GribiNhAftVersionType . . . 1449
2.1829 show traffic-engineering gribi nexthop-group-aft [...] . . . 1450
2.1829.1 GribiNhgAftType . . . 1450
2.1830 show traffic-engineering gribi nexthop-group-aft unacknowledged [...] . . . 1451
2.1830.1 GribiNhgAftVersionType . . . 1451
2.1831 show traffic-engineering interfaces [...] . . . 1452
2.1831.1 TeIntfModel . . . 1452
2.1832 show traffic-engineering interfaces bandwidth [...] . . . 1453
2.1832.1 TeInterfaceBwModel . . . 1453
2.1833 show traffic-engineering rsvp . . . 1453
2.1834 show traffic-engineering rsvp tunnel [...] . . . 1455
2.1834.1 RsvpLerTunnel . . . 1455
2.1835 show traffic-engineering segment-routing cbf fec [...] . . . 1458
2.1835.1 SrTePolicyCbfFecModel . . . 1458
2.1836 show traffic-engineering segment-routing policy [...] . . . 1459
2.1836.1 SrTePolicyCandidatesModel . . . 1459
2.1837 show traffic-engineering segment-routing policy active [...] . . . 1461
2.1837.1 SrTePolicyActiveModel . . . 1461
2.1838 show traffic-engineering segment-routing policy fec [...] . . . 1462
2.1838.1 SrTePolicyFecInfo . . . 1462
2.1839 show traffic-engineering segment-routing policy segment-list [...] . . . 1462
2.1839.1 SrTePolicySegmentLists . . . 1463
2.1840 show traffic-engineering segment-routing policy segment-list counters [...] . . . 1463
2.1840.1 SrTePolicySegmentListsCounters . . . 1463
2.1841 show traffic-engineering segment-routing policy summary . . . 1464

2.1841.1 SrTePolicySummaryModel . . . 1464
2.1842 show traffic-engineering segment-routing replication-segment [...] . . . 1466
2.1842.1 SrP2mpCandidatesModel . . . 1466
2.1843 show traffic-engineering segment-routing replication-segment summary . . . 1468
2.1843.1 ReplicationSegmentSummaryModel . . . 1468
2.1844 show traffic-policy (cpu | vlan | interface) [...] . . . 1470
2.1844.1 TrafficPolicy . . . 1470
2.1845 show traffic-policy (cpu | vlan | interface) counters [...] . . . 1471
2.1845.1 TrafficPolicyCounters . . . 1471
2.1846 show traffic-policy (cpu | vlan | interface) errors [...] . . . 1472
2.1846.1 TrafficPolicyErrors . . . 1472
2.1847 show traffic-policy (cpu | vlan | interface) summary [...] . . . 1474
2.1847.1 TrafficPolicySummary . . . 1474
2.1848 show traffic-policy field-set integer [...] . . . 1475
2.1848.1 TrafficPolicyIntegerFieldSet . . . 1475
2.1849 show traffic-policy field-set ipv4 prefix [...] . . . 1475
2.1849.1 IpPrefixFieldSet . . . 1475
2.1850 show traffic-policy field-set ipv6 prefix [...] . . . 1476
2.1851 show traffic-policy field-set l4-port [...] . . . 1476
2.1851.1 TrafficPolicyL4PortFieldSet . . . 1476
2.1852 show traffic-policy field-set mac [...] . . . 1476
2.1852.1 TrafficPolicyMacFieldSet . . . 1476
2.1853 show traffic-policy field-set vlan [...] . . . 1476
2.1853.1 TrafficPolicyVlanFieldSet . . . 1477
2.1854 show traffic-policy protocol neighbors bgp . . . 1477
2.1854.1 ProtocolNeighbor . . . 1477
2.1855 show transceiver application [...] . . . 1477
2.1855.1 TransceiverApplications . . . 1477
2.1856 show transceiver error [...] . . . 1478
2.1856.1 TransceiverErrorValues . . . 1478
2.1857 show transceiver error-correction coherent degrade [...] . . . 1479
2.1857.1 TransceiverCoherentFecDegradeSlot . . . 1479
2.1858 show transceiver squelch [...] . . . 1480
2.1858.1 TransceiverSquelchTypes . . . 1480
2.1859 show transceiver status [...] . . . 1480
2.1860 show transceiver status firmware update [...] . . . 1481
2.1860.1 TransceiverFirmwareUpdateSlotDetail . . . 1481
2.1861 show transceiver status firmware version [...] . . . 1482
2.1861.1 TransceiverFirmwareCollection . . . 1482
2.1862 show transceiver traffic-loopback capabilities [...] . . . 1482
2.1862.1 Capabilities . . . 1482
2.1863 show transceiver traffic-loopback status [...] . . . 1483
2.1863.1 SlotInfo . . . 1483
2.1864 show tunnel counters ip-in-ip [...] . . . 1483
2.1864.1 TunnelCountersEntry . . . 1483
2.1865 show tunnel counters mpls rate [...] . . . 1484
2.1865.1 MplsTunnelCountersRateEntry . . . 1484
2.1866 show tunnel fib [...] . . . 1484
2.1866.1 TunnelFibCategory . . . 1484
2.1867 show tunnel fib multicast [...] . . . 1485
2.1868 show tunnel rib (system-tunneling-ldp-tunnel-rib | system-igp-shortcut-tunnel-rib | system-tunnel-rib | TUNNEL_RIB_NAME) summary . . . 1485
2.1869 show tunnel rib brief [...] . . . 1486
2.1869.1 TunnelRibEntriesByTunnelType . . . 1486
2.1870 show tunnel rib candidates [...] . . . 1487

2.1870.1 InternalTunnelRibCandidates . . . 1487
2.1871 show tunnel rib colored system-colored-tunnel-rib brief [...] . . . 1487
2.1871.1 TunnelRibColors . . . 1487
2.1872 show tunnel rib multicast (mldp | static) . . . 1488
2.1872.1 MulticastTunnelRibEntry . . . 1488
2.1873 show tunnel rib summary . . . 1488
2.1874 show upnp counters [...] . . . 1488
2.1874.1 VrfCounters . . . 1488
2.1875 show upnp igd port-map . . . 1489
2.1875.1 VrfUpnpPortMap . . . 1489
2.1876 show upnp service . . . 1489
2.1876.1 UpnpServiceEntry . . . 1489
2.1877 show uptime . . . 1489
2.1878 show user-account . . . 1490
2.1878.1 User . . . 1490
2.1879 show users . . . 1490
2.1879.1 ShowAaaUserInfo . . . 1490
2.1880 show users accounts . . . 1490
2.1881 show users detail . . . 1490
2.1882 show users roles [...] . . . 1490
2.1883 show version [...] . . . 1491
2.1883.1 Details . . . 1491
2.1884 show version compatibility SWI . . . 1492
2.1885 show version firmware . . . 1492
2.1886 show version license . . . 1492
2.1887 show vlan [...] . . . 1492
2.1887.1 Status . . . 1493
2.1888 show vlan brief [...] . . . 1493
2.1889 show vlan brief count . . . 1493
2.1890 show vlan counters [...] . . . 1493
2.1890.1 VlanStats . . . 1493
2.1891 show vlan dynamic . . . 1494
2.1891.1 VlanIds . . . 1494
2.1892 show vlan interface dynamic . . . 1494
2.1892.1 VlanIntfs . . . 1494
2.1893 show vlan internal allocation policy . . . 1494
2.1894 show vlan internal usage . . . 1494
2.1895 show vlan internal usage reserved . . . 1494
2.1896 show vlan private-vlan . . . 1495
2.1896.1 PrivateVlans . . . 1495
2.1897 show vlan summary . . . 1495
2.1898 show vlan trunk group [...] . . . 1495
2.1898.1 GroupNames . . . 1495
2.1899 show vmtracer all [...] . . . 1496
2.1899.1 VmTracerSwitches . . . 1496
2.1900 show vmtracer interface (ETH | ETHS | LAG) vm VMNAME . . . 1497
2.1900.1 VmTracerGenericInterface . . . 1497
2.1900.2 VmTracerLagInterface . . . 1497
2.1901 show vmtracer interface [...] . . . 1497
2.1902 show vmtracer logical-switch [...] . . . 1497
2.1902.1 NetworkScope . . . 1498
2.1903 show vmtracer session SESSION (vcenter | vsm) [...] . . . 1499
2.1903.1 VmTracerSessionStatus . . . 1499
2.1904 show vmtracer session [...] . . . 1500
2.1905 show vmtracer vm VM_NAME . . . 1500

2.1906 show vmtracer vm [...] . . . 1500
2.1907 show vmtracer vxlan segment [...] . . . 1500
2.1908 show vmtracer vxlan segment range . . . 1500
2.1908.1 McastRange . . . 1500
2.1908.2 VniRange . . . 1501
2.1909 show vpls [...] . . . 1502
2.1909.1 VplsInstance . . . 1502
2.1910 show vpls pseudowire brief (all | { (neighbor NEIGHBOR) | (pseudowire-type PW_TYPE) }) [...] . . . 1504
2.1911 show vrf [...] . . . 1504
2.1911.1 VrfEntry . . . 1504
2.1912 show vrf leak connected (ipv4 | ipv6) . . . 1505
2.1912.1 LeakedConnectedRoutes . . . 1505
2.1913 show vrf leak id . . . 1505
2.1914 show vrf reserved-ports [...] . . . 1505
2.1914.1 VrfReservedPortsEntry . . . 1505
2.1915 show vrf selection policy . . . 1505
2.1916 show vrf summary . . . 1505
2.1917 show vrrp [...] . . . 1506
2.1917.1 VrrpRouterDetailBase . . . 1506
2.1917.2 VrrpRouterDetailV4 . . . 1507
2.1917.3 VrrpRouterDetailV6 . . . 1508
2.1918 show vrrp brief [...] . . . 1509
2.1918.1 VrrpRouterBriefBase . . . 1509
2.1918.2 VrrpRouterBriefV4 . . . 1509
2.1918.3 VrrpRouterBriefV6 . . . 1509
2.1919 show vxlan address-table [...] . . . 1510
2.1919.1 VxlanMacAddrListing . . . 1510
2.1920 show vxlan address-table count [...] . . . 1510
2.1921 show vxlan config-sanity [...] . . . 1510
2.1921.1 ConfigCheckCategory . . . 1510
2.1922 show vxlan control-plane [...] . . . 1511
2.1922.1 VlanControlPlaneConfigModel . . . 1511
2.1923 show vxlan controller address-table advertised [...] . . . 1511
2.1924 show vxlan controller address-table received [...] . . . 1511
2.1925 show vxlan controller arp advertised [...] . . . 1512
2.1926 show vxlan controller arp received [...] . . . 1512
2.1927 show vxlan controller nd advertised [...] . . . 1512
2.1928 show vxlan controller nd received [...] . . . 1512
2.1929 show vxlan controller status . . . 1512
2.1930 show vxlan counters learn-restrict COUNTER [...] . . . 1512
2.1930.1 VxlanPrefixStatusModel . . . 1512
2.1931 show vxlan counters software . . . 1514
2.1932 show vxlan counters varp . . . 1519
2.1933 show vxlan counters vni [...] . . . 1523
2.1933.1 VxlanVniCounters . . . 1523
2.1934 show vxlan counters vtep [...] . . . 1523
2.1934.1 VxlanVtepCounters . . . 1523
2.1935 show vxlan flood traffic . . . 1524
2.1936 show vxlan flood vtep [...] . . . 1524
2.1936.1 VxlanVtepListModel . . . 1524
2.1937 show vxlan learn-restrict vtep [...] . . . 1524
2.1937.1 VxlanPrefixListModel . . . 1524
2.1938 show vxlan policers vni [...] . . . 1525
2.1938.1 VxlanConfigModel . . . 1525

2.1939 show vxlan policers vni counters [...] . . . 1525
2.1939.1 VxlanVniPolicersAllCountersInfoModel . . . 1526
2.1940 show vxlan qos . . . 1526
2.1941 show vxlan security profile . . . 1526
2.1942 show vxlan vni [...] . . . 1526
2.1942.1 VxlanVniVlanCollection . . . 1527
2.1943 show vxlan vni summary . . . 1527
2.1944 show vxlan vtep [...] . . . 1527
2.1944.1 RemoteVtepConfig . . . 1528
2.1944.2 TunnelTypeList . . . 1528
2.1944.3 VxlanInterface . . . 1528
2.1945 show vxlan vtep summary . . . 1528
2.1946 show world . . . 1528
2.1946.1 Country . . . 1528
2.1947 show world bad . . . 1529
2.1948 show world stream [...] . . . 1529
2.1949 show xmpp neighbors . . . 1529
2.1949.1 Neighbor . . . 1529
2.1950 show xmpp status . . . 1529
2.1951 show xmpp switch-group . . . 1530
2.1951.1 Group . . . 1530
2.1952 show zerotouch . . . 1530
2.1953 traceroute mpls generic ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...] . . . 1531
2.1953.1 MplsTracerouteHdr . . . 1531
2.1954 traceroute mpls ldp ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...] 1534
2.1955 traceroute mpls p2mp mldp ROOTADDR ((lsp-id OPQVAL) | (multicast ipv4 SOURCEADDR GROUPADDR)) [...] . . . 1534
2.1956 traceroute mpls rsvp session ((id SESSION_ID lsp LSP) | (name SESSION_NAME)) [...] 1534
2.1957 traceroute mpls rsvp tunnel TUNNEL_NAME [...] . . . 1534
2.1958 traceroute mpls segment-routing ((((ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (ipv6 IPV6_ADDRESS) | (isis ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (isis ipv6 IPV6_ADDRESS))) | (ospf ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)))) [...] . . . 1534

Chapter 1

Arista Command API Fundamentals

The Arista Command API is a simple and complete API that allows you to configure and monitor your Arista switches. Once the API is enabled, the switch accepts commands using the industry standard CLI syntax, and responds with machine readable output and errors serialized in JSON, served over HTTP.

The Command API is easy to use...

```
from jsonrpclib import Server

switch = Server( "https://username:passw0rd@myswitch/command-api" )
response = switch.runCmds( 1, ["show version"] )
print "The switch's system MAC addess is", response[0]["systemMacAddress"]
```

... and easy to configure!

```
bash$ ssh username@myswitch
Password:  ⟨passw0rd⟩
myswitch⟩ enable
myswitch# configure terminal
myswitch(config)# management api http-commands
myswitch(config-mgmt-api-http-cmds)# no shutdown
myswitch(config-mgmt-api-http-cmds)# show management api http-commands
Enabled:            Yes
HTTPS server:       running, set to use port 443
HTTP server:        shutdown, set to use port 80
Local HTTP server:  shutdown, no authentication, set to use port 8080
Unix Socket server: shutdown, no authentication
VRFs:               default
...
```

1.1 Configuring the Command API Agent

Although disabled by default, it is very simple to get the Command API server running on your switch. To modify its configuration, from `configure` mode, enter `management api http-commands` mode. In this submode, you can turn on or off the server by typing `[no] shutdown`.
The configuration of the protocol itself in done in HttpServer mode. From `configure` mode, enter `management http-server` mode. In this submode, you can switch between accepting HTTP or HTTPS traffic via `[no] protocol http[s]`, and adjust the ports the server should listen on using `protocol http[s] port` ⟨portNumber⟩.
You can also configure the API to only accept connections from scripts running on the switch itself. To enable this feature, see 1.4.
In any CLI mode, you can issue the `show management api http-commands` command to view the status of the agent, including server information, attached users, last request time and other details.

```
myswitch⟩ enable
myswitch# configure terminal
myswitch(config)# management api http-commands
myswitch(config-mgmt-api-http-cmds)# [no] shutdown
myswitch(config-mgmt-api-http-cmds)# management http-server
myswitch(config-mgmt-http-server)# [no] protocol https [port <portNumber>]
myswitch(config-mgmt-http-server)# [no] protocol http [port <portNumber>]
myswitch(config-mgmt-http-server)# [no] protocol http localhost [port <portNumber>]
myswitch(config-mgmt-http-server)# [no] protocol unix-socket
```

1.1.1 Configuring a Certificate

Because clients use HTTP basic authentication to send usernames and passwords to the switch, we recommend using HTTPS so no passwords are sent in the clear over the network. HTTPS certificates and key can be changed with `[no] protocol https certificate`. By default a self-signed certificate will be used.

```
myswitch⟩ enable
myswitch# configure terminal
myswitch(config)# management http-server
myswitch(config-mgmt-http-server)# protocol https certificate
Enter TEXT certificate.  Type 'EOF' on its own line to end.
...
EOF

Enter TEXT private key.  Type 'EOF' on its own line to end.
...
EOF
```

1.1.2 Configuring multiple VRFs

Since release 4.20, you can configure multiple VRFs in `management api http-commands` mode. By entering submode `vrf` ⟨vrfName⟩, you can enable or disable Command API server in multiple VRFs by typing `[no] shutdown` inside each VRF submode. If no VRF specified and Command API server is enabled, it will start in default VRF automatically. When you enable Command API server in the first user VRF, it will be removed from default VRF. You can run Command API server in default VRF by explicitly configuring in submode `vrf default` with `no shutdown` (this behavior is for reasons of backwards compatibility).

```
myswitch⟩ enable
myswitch# configure terminal
myswitch(config)# management api http-commands
myswitch(config-mgmt-api-http-cmds)# vrf foo
myswitch(config-mgmt-api-http-cmds-vrf-foo)# no shutdown
myswitch(config-mgmt-api-http-cmds-vrf-foo)# exit
myswitch(config-mgmt-api-http-cmds)# vrf bar
myswitch(config-mgmt-api-http-cmds-vrf-bar)# no shutdown
myswitch(config-mgmt-api-http-cmds-vrf-bar)# exit
myswitch(config-mgmt-api-http-cmds)# vrf default
myswitch(config-mgmt-api-http-cmds-vrf-default)# no shutdown
myswitch(config-mgmt-api-http-cmds-vrf-default)# exit
myswitch(config-mgmt-api-http-cmds)# show management api http-commands
Enabled:            Yes
HTTPS server:       running, set to use port 443
HTTP server:        shutdown, set to use port 80
Local HTTP server:  shutdown, no authentication, set to use port 8080
Unix Socket server: shutdown, no authentication
VRFs:               bar,default,foo
...
```

1.2 The Spec

The Command API uses the lightweight, standardized JSON-RPC protocol to communicate between your program (the client) and the switch (the server), see http://www.jsonrpc.org/specification. Although most languages have client libraries which abstract away the JSON-RPC internals, it is useful to understand the communication mechanism between the client and the switch.

1.2.1 Exploring the Command API

To explore the API, point your web browser to `http[s]://myswitch/`, after enabling Command API. This web-app lets you interactively explore the protocol, return values, and model documentation. Please note that if you visit the tool using HTTPS, your browser may emit a warning; this is because by default the agent uses a locally signed certificate for encryption purposes and can be ignored. You may always set a certificate and key issued by your own CA for enhanced security.

1.2.2 An annotated example

The client starts by sending a JSON-RPC request via an HTTP POST request to http://⟨your-switch⟩/command-api, which encapsulates a list of CLI commands it wishes to run, and the switch replies with a JSON-RPC response containing the result of each CLI command that was executed. The commands in the request are run in order on the switch. After the switch has executed all commands, it exits back to unprivileged mode. If any of the commands emit an error, no further commands from that request are executed, and the response from the switch will contain an `error` object containing the details of the error that occurred.

To illustrate, here are annotated versions of the underlying Request and Response messages sent between the client and switch.

1.2.2.1 The Request

```
{
 "jsonrpc": "2.0",
 "method": "runCmds",
 "params": {
   "version": 1,
   "cmds": [
     "enable",
     "configure",
     "interface Ethernet1",
     "switchport access vlan 3",
     "show vlan 3",
   ],
   "format": "json",
 },
 "id": 1
}
```

`jsonrpc`: The protocol version. Always `2.0`, and usually set by a client library.

`method`: The method to run on the switch. At this time `runCmds` is the only method supported.

`version` is a mandatory parameter representing the API version. Currently always has the value `1`.

`cmds` is a mandatory parameter which contains a list of CLI commands. Commands may either be a string or a complex object type. For more details, see the command specification

`format` is an optional parameter specifying response format. Can be `text` or `json`, and defaults to `json`.

`id`: A unique identifier that will be echoed back by the switch. May be a string or number.

1.2.2.2 The Response: success case

This response is returned by the switch after receiving the above request. All responses conform JSON-RPC 2.0 specifications.

```
{
 "jsonrpc": "2.0",
 "result": [
   {},
   {},
   {},
   { "warnings": [
       "Access VLAN does not exist.  Creating vlan 3"
     ]
   },
   { "sourceDetail": "",
     "vlans": {
       "3": {
         "status": "active",
         "name": "VLAN0003",
         "interfaces": {
           "Ethernet1": {
              "annotation": null,
              "privatePromoted": false
           }
         },
         "dynamic": false
       }
     }
   }
 ],
 "id": 1
}
```

The protocol version. Always `"2.0"`.

The result list. Contains one object for each CLI command executed, and the position of the result corresponds to the original command list. If the request specified a `text` format, however, `result` will contain a list of strings.

Most non-show commands produce no output and thus return an empty JSON object. However, in some cases, a command may produce `warnings` or `errors` which will be stored in arrays of the same name.

show commands will return formatted JSON objects containing all information seen in the ASCII output in the normal CLI. In this case, `show vlan 3` will give you a mapping from VLAN id to VLAN model.

The same id (string/number) that was provided by the request.

1.2.2.3 The Response: failure case

In this case, let's assume our account does not have the appropriate AAA permissions to enter `configure` mode.

```
{
   "jsonrpc": "2.0",
   "error": {
```

On an `error`, no result object is present, only an error object, which is guaranteed to have the following attributes: `code`, `messages`, and `data`.

```
     "code": 1005,
```

Command API `error code`.

```
     "message": "CLI command 2 of 5 'configure' failed:
                 permission to run command denied",
```

General high-level error message.

```
     "data": [
```

Similar to the `result` object in the successful response, the `data` object is a list of objects corresponding to the results of all commands up to, and including, the failed command. If there was a an error before any commands were executed (e.g. bad credentials), `data`, will be empty.

```
         {},
```

The `enable` command passed without incident.

```
         { "errors": [
           "Authorization denied for command 'configure'"
         ] }
     ]
   },
   "id": 1
}
```

In this case, we failed on the second command. The last object in the `data` array will always correspond to the failed command. The command failure details are always stored in the `errors` array.

1.2.3 Command Specification

In most cases, the client will use a simple string to specify the CLI command in the `cmds` parameter in the request. In certain cases, however, clients may wish to specify additional parameters during the command's execution. To use complex commands, pass a JSON object in lieu of a string, with the following attributes:

`cmd`	(mandatory) specify the CLI command to run.
`input`	(optional) specify a string to be provided as standard input while running the `cmd`
`revision`	(optional) in the case of 'show' commands that have been modified over the course of different EOS releases, this parameter allows clients to request an old model format. At this time, all models are at revision 1, and this attribute will be ignored.

For example, to set the message of the day to `Hello World!`, the client should set `cmds` to

```
[ "enable", "configure", { "cmd":  "banner motd", "input":  "Hello World!\nEOF" } ]
```

Similarly, if the switch requires an enable password, the following `cmds` value would let you enter exec mode and clear interface counters

```
[  "cmd":  "enable", "input":  "hunter2" , "clear counters" ]
```

1.2.4 Error Codes

The following errors are generated if there is an issue with the request. Do note that new error codes may be introduced in future releases.

JSON-RPC 2.0 Defined Errors

Code	Description
32700	Parse Error
32600	Invalid Request
32601	Method not found
32602	Invalid params
32603	Internal server error

Command API Errors

Code	Description
1000	General error
1001	Command caused internal exception
1002	Invalid command: the string does not match any command
1003	Command not ready: attempted to get json from a show command, however the command has not been converted to be compatible with the Command API yet. Use format = text to run this command and get ASCII output.
1004	Command incompatible: cannot run command over the Command API
1005	Command unauthorized: user has insufficient permissions to run the command
1006	Configuration locked: another CLI session has the configuration lock

1.3 Using the Command API

In practice, most libraries hide the JSON-RPC version, id and protocol details. Instead you only need to call the `runCmds` method with the `version` and `cmds` parameters and, optionally, the `format` parameter. The responses generated by the client library usually follow language conventions. For example, in Python, an error response results in

an Exception being thrown, while Javascript expects an error handler callback.

1.3.1 Unsupported commands

Certain commands are not permitted and will always return an error. The largest class of such commands are interactive commands; `watch` is a prime example. Other commands, like `reload` must use their non-interactive versions: such as `reload now`. The `bash` command is only allowed with the `timeout <seconds>` argument, ie. `bash timeout <seconds> <bash-command>`.

Commands that attempt to use CLI pipes (e.g. `show interfaces | grep Ethernet1`) are also not allowed. In addition, no abbreviations are allowed in commands. This is necessary because future versions of EOS may add more commands, rendering previous abbreviations ambiguous.

1.3.2 Unconverted commands

Although you can access almost any CLI command via the Command API, not all `show` commands have been converted to return formatted data, and trying to run the command with the `format` parameter set to `json` will result in an error. However, you can still get the CLI ASCII output for the unconverted command by setting the `format` parameter to `text`. With this parameter set, the response will contain an array of strings (rather than an array of objects), each representing the text output of the corresponding command. When operating with `format` set to `text`, no errors will be generated; rather they will be included in the ASCII output, the same way they are printed in a standard CLI session. To view converted commands visit the `documentation.html` on a switch.

1.3.3 Streamed commands

eAPI maintains a buffer that can be exhausted if the return value is too large. eAPI offers a special streaming mode that does not buffer any of the results. The benefits of using streaming is it uses less memory and less CPU. However errors are not handled in the standard JSON-RPC fashion. According to the JSON-RPC spec an `error` object must appear at the top level of the response. However in the streaming case the `error` object appears within the `result` object. Additionally valid JSON may not be produced in an error case. To use streaming please use `streaming=true`

1.3.4 Using JSON-RPC

The JSON-RPC spec is well documented at `http://www.jsonrpc.org/specification`, and there are libraries in almost every language that make dealing with JSON-RPC a breeze. Here are a list of some recommended client libraries:

Python : `https://github.com/joshmarshall/jsonrpclib`

Perl : `http://search.cpan.org/~powerman/JSON-RPC2-0.1.1/lib/JSON/RPC2/Client.pm`

Javascript : `https://github.com/gimmi/jsonrpcjs`

And many more `http://en.wikipedia.org/wiki/JSON-RPC#Implementations`

1.4 On-box Usage of Command API

It is often useful to run scripts that use Command API directly on the switch itself, without configuring them with AAA credentials. EOS provides three mechanisms to accomplish this. The preferred way is to use the EapiClientLib library, however other methods are also supported.

1.4.1 Using EapiClientLib (requires no configuration)

```
from EapiClientLib import EapiClient

switch = Server( disableAaa=True, privLevel=15 )
response = switch.runCmds( 1, ["show version"] )
print "The switch's system MAC addess is", response["result"][0]["systemMacAddress"]
```

1.4.2 Other Methods (requires configuration)

The first is an HTTP server bound to localhost (on port 8080 by default), which only accepts connections arriving from the same machine. The other solution is a Unix domain socket. To enable one or both, use the following commands:

```
myswitch⟩ enable
myswitch# configure terminal
myswitch(config)# management api http-commands
myswitch(config-mgmt-api-http-cmds)# protocol http localhost
myswitch(config-mgmt-api-http-cmds)# protocol unix-socket
myswitch(config-mgmt-api-http-cmds)# no shutdown
```

Once Command API is enabled then you access via the local domain socket `unix:/var/run/command-api.sock`

```
switch = Server( "unix:/var/run/command-api.sock" )
```

If configured to use HTTP over localhost, your script can access the API as follows:

```
switch = Server( "http://localhost:8080/command-api" )
```

1.5 Using Websockets with Command API

Command API can also be used over Websockets, allowing for a stateful approach, which jsonrpc does not provide on its own. Commands executed over Websockets do not create a new session for each request. One use case is if multiple requests needs to be performed while holding the configuration lock. There is a reference implementation of the protocol in the python site-packages named EapiWebsocketLib.py. This is an example usage if it were done on-box:

```
import websocket
from EapiWebsocketLib import EapiWebsocketClient

ws = websocket.WebSocket()
with EapiWebsocketClient( ws, "127.0.0.1", username, password ) as switch:
    response = switch.runCmds( 1, ["show version"] )
    print "The switch's system MAC addess is", response["result"][0]["systemMacAddress"]
```

1.6 Client-Side Certificates

Client authentication can be accomplished via X.509 public key certificates.

What is this good for? Imagine you have a general purpose script that does some operations on an Arista switch using eAPI. That script will need your EOS username/password. Since the various users of that script might have their own EOS usernames/passwords, putting them into the script makes little sense; having the script collect them from a well-known password file relative to $HOME makes more sense. The public key certificates is simply the standard way of dealing with such a well-known password file.

Public key certificates are digitally signed by a Certificate Authority (CA). If you trust that CA, then you must trust the certificates it signed. The trusted CA(s) has to be configured into the Arista switch. Then any client certificate signed by that trusted CA will be accepted by eAPI, but only if the CA directly signed it, that is, a client cannot create "sub-clients" (the CA plays the role of the admin that creates the account for which the certificate stands). Client certificates can also be self-signed, in which case each client certificate has to be configured into the Arista switch as trusted.

The certificate will need to contain the username and its privilege level (a number between 1 and 15) in the "common name" field of the certificate's "subject" field (see openssl command syntax below for more exact details). If a connection with this certificate is accepted, then on the switch it will be accounted against that username and given specified privilege level. Note that this will bypass RADIUS/TACACS+.

If you configure trusted CA/clients, then you also need to configure the HTTPS server's certificate/privateKey using the `management security ssl profile` commands as illustrated below (server keys are no longer auto-generated in this case – once you generate the client's certificate, you might as well generate the server certificate and key). Those certificates/keys are stored in bootflash under `/persist/secure/ssl`, not in the startup-config.

```
myswitch⟩ enable
myswitch# configure
myswitch(config)# !  download the client certificates and server certificate/key
myswitch(config)# copy scp:<username>@<server>//<certificate1-filename> certificate:ca1
myswitch(config)# copy scp:<username>@<server>//<certificate2-filename> certificate:user1
myswitch(config)# copy scp:<username>@<server>//<certificate3-filename> certificate:httpServer
myswitch(config)# copy scp:<username>@<server>//<privateKey3-filename> sslkey:httpServer
myswitch(config)# !  create an ssl profile and add those certs/keys into it
myswitch(config)# management security
myswitch(config-mgmt-security)# ssl profile capi
myswitch(config-mgmt-sec-ssl-profile-eapi)# certificate httpServer key httpServer
myswitch(config-mgmt-sec-ssl-profile-eapi)# trust certificate ca1
myswitch(config-mgmt-sec-ssl-profile-eapi)# trust certificate user1
myswitch(config)# !  now configure eAPI to use that ssl profile
myswitch(config)# management api http-commands
myswitch(config-mgmt-api-http-cmds)# protocol https
myswitch(config-mgmt-api-http-cmds)# protocol https ssl profile capi
myswitch# !  some show commands
myswitch# show management security ssl certificate
myswitch# show management security ssl profile
myswitch# show management security ssl key
myswitch# !  find more commands with:
myswitch# show cli commands all | grep ssl
```

Above we first downloaded certificates and a private key (key is for the https server). Certificates (that contain the public key) go to the `certificate:` filesystem, (private) keys go to the `sslkey:` filesystem. Then we add those certificates/keys into an ssl profile called `capi`, then make command-api use that profile.

A self signed certificate and associated private key (a "pair") can be generated with the `openssl` tool like so: (re-

place the ⟨fields⟩ with actual values)

```
openssl req -new -nodes -x509 -days <number-of-days>
   -subj '/CN=<username> <privilege> <optional-stuff>'
   -out <certificate-filename>
   -keyout <key-filename>
```

You can use the above for generating the `certificate:httpServer` and `sslkey:httpServer` or the `certificate:user1/ca1` used in the example above. In case of the client/user, the key-file is for the client to install in his account for access by tools like wget/browsers (in the same way as a username/password would be). In case of the CA, you (the admin), would keep that key to sign client certificates with.

You can create a self signed certificate per user, and have EOS trust each one, or you can create a CA, have EOS trust just that CA, then sign the user certificates you generate with the (private) key of the CA. In that case, creating the certificate is a 2 step process: you first create a "signing-request" (similar to creating a cert/key pair, but without the "-x509 -days" options), which is then signed in the second step.

```
openssl req -new -nodes
   -subj '/CN=<username> <privilege>'
   -out <signing-request-filename>
   -keyout <user-key-filename>

openssl x509 -req -sha256 -days <valid-for-days>
   -in <signing-request-filename>
   -CA <ca-certificate-filename> -CAkey <ca-key-filename>
   -set_serial <serial-number>
   -out <user-certificate-filename>
```

An eAPI request can be generated like so by using the wget utility: (the –no-check-certificate is required if the server uses a self-signed certificate)

```
wget -no-check-certificate
   -private-key=<key-filename>
   -certificate=<certificate-filename>
   -q -O - https://<server>/command-api -post-data="$(cat <file-with-json-request>)"
```

A simple `<file-with-json-request>` could have this content:

```
{
   "jsonrpc": "2.0",
   "method": "runCmds",
   "params": {
     "version": 1,
     "cmds": [ "show clock" ],
     "format": "json"
   },
   "id": "1"
}
```

1.7 Session Based Authorization

Some systems use 2-factor authentication where RSA is implemented and keys rotate every 30 seconds. It is cumbersome to authenticate using username and password in these systems. Command API requests use basic authentication by default. This results in high overhead on the switch to verify user credentials for each command request. HTTP sessions can be used to reduce the overhead in the above use cases by caching user authentication. Usage of the session mechanism is explained below.

1.7.1 User Login

A user logs in by sending an HTTP POST request to `http[s]://<yourswitch>/login` with an encapsulated JSON user credential.

```
{
   "username":  <user>,
   "password":  <password>,
}
```

The switch will authenticate the user credentials in the request and if successful, send a set cookie response containing a `Session=<sessionId>` field and an expiry of 24 hours. The sessionId is an unique session identifier. On failure, a "401 Unauthorized" HTTP response will be sent in lieu of a cookie.

1.7.2 Command Requests

Subsequent command requests to `http[s]://<yourswitch>/command-api` need to include `Session=<sessionId>` in the cookie header. The format of the command request as well as its response is same as the one using the basic authentication scheme. If the session is invalid, a "401 Unauthorized" response will be sent.

1.7.3 User Logout

A user may logout by sending an HTTP POST request to `http[s]://<yourswitch>/logout` with `Session=<sessionId>` in the cookie header. On success, a "200 OK" response will be sent, otherwise a "400 Bad Request" response will be sent.

1.7.4 Session Validation

Browser-based applications may find it useful to check if a valid session already exists, thereby avoiding a prompt. To do this, send a POST request to `http[s]://<yourswitch>/login` with an empty JSON body: `{}`. If the cookie included in your request exists and contains a valid sessionId, the server will send a "200 OK" HTTP response, otherwise it will reply with a "401 Unauthorized" response.

1.8 Versioning

eAPI models are now revisioned.
This means that if a model is modified in a non-backwards compatible way, then its revision will be bumped up (revisions are numbers, default value is 1).
By default an eAPI request will return revision 1 of the model instance, this ensures that older management software will not suddenly stop working when a switch is upgraded.
If a specific revision is required, then that revision must be specified in the json rpc, like illustrated in this example json request below:

```
{
  "jsonrpc": "2.0",
  "method": "runCmds",
  "params": {
    "version": 1,
    "cmds": [
      "enable",
      { "cmd": "show aliases", "revision": 2 },
      { "cmd": "show interfaces", "revision": 3 },
      "show clock"
    ],
   "format": "json"
  },
  "id": "1"
}
```

1.8.1 Incompatible Model

The following list contains changes that will result in a new revision of a model. In the list below, the term "element" refers to an attribute or submodel.

- Changing the type/format of an element (say, from integer to string)
- Removing mandatory elements
- Adding mandatory elements
- Changing the optionality of an element

Looking at it from the other side, the model changes that can happen without a revision bump can be summed up as:

- Changing help text
- Using a stronger type (from string to any, i.e. number, is compatible)
- Adding/removing optional elements
- Adding/removing an enumerated value
- Obviously: adding a new cli command

Normally, in TLV/schema-less systems (like JSON is), adding new key-value pairs is considered a compatible change, as long as the new pairs are just an "enhancement" that does not affect the meaning of the remaining pairs, since older system will be oblivious to them (they naturally skip over them).
So why is adding a mandatory element non-backwards compatible? Because we are talking from the perspective of the published model.
If a network management developer looks at the latest model and sees that model X is at revision 1 and attribute Y is mandatory, then he will assume Y to be always present, then get a key error when discovering a switch running an older release! The downside is that you need extra effort to get the newer attributes (specify a later revision), but

unless you add code to deal with that extra data, you might as well just continue receiving only the older attributes. If your application can deal with "fuzzy models", then look for "Get Current Revision" further below.

1.8.2 Understanding Revisions And Versions

Do not confuse "revisions" and "versions" as they are not the same. A "revision" applies to a particular CLI command whereas a "version" is global and is internally translated to a specific "revision" for each CLI command in the rpc. At this time, the API remains at version 1, and will do so until many models have progressed to later revisions. Until the version gets a bump, in order to access newer revisions of CLI output, you have to name the revision you desire, per command, or use the "latest" version as documented in the "Accessing a Command's Current Revision" section below. Note that an explicit revision value attached to a command always takes precedence over the version.

Here is how revisions and versions play together.
Assume commands C1 and C2 have shipped with those changes across EOS releases (C3 is new and has no revisions yet – still at revision=1):

Versioning Examples

Command	EOS-Release	eAPI-Version	Command-Revision
C1	4.13.1	1	1
C1	4.13.2	2	2
C1	4.13.3	2	3
C1	4.13.4	3	3
C1	4.13.5	3	4
C2	4.13.1	1	1
C2	4.13.4	3	2
C3	4.13.5	1	1

Then, if querying a switch running Eos-release 4.13.5, we get:

Returned Models

Command	Version-Requested	Revision-Returned
C1	1	1
C1	2	2
C1	3-*	3
C2	1-2	1
C2	3-*	2
C3	1-*	1

If a revision bump occurs without version bump, then, for a given version, you are stuck at the initial revision level: no matter the version, you will never get revision 4 for C1. The only way to see a more up-to-date output (for example C1 at revision 4) is to explicitely ask for that revision of the command, or to request the current revision (see next chapter), which could return anything from the distant future, thus unpredictable.

Note that requiring a version higher than the command knows about will simply return the latest available one.

1.8.3 Accessing a Command's Current Revision

If your code can deal with new output naturally, then you may want to always get output according to the latest revision a switch can provide (its current revision), which is possible by providing the special version value 'latest' (in that case only, the version is a string). For example:

```
{
  "jsonrpc": "2.0",
  "method": "runCmds",
  "params": {
     "version": "latest",
     "cmds": [
        "enable",
        "show ip interfaces",
     ],
     "format": "json"
  },
  "id": "1"
}
```

1.8.4 More Documentation On Revisions

Since there is now infrastructure in place during the software development cycle to detect changes to models and force the bumping up of revision numbers, thus ensuring that eAPI is a stable and predictable API, we eventually could also publish our per revision âĂIJReference ModelsâĂİ, as those are a python structures and thus more formally usable than a web-page or pdf.

Chapter 2

Command Output

All commands are guaranteed to have structured response format, documented here. If an attribute is marked as optional (name enclosed in `[]` brackets), its presence is not guaranteed in the resulting JSON respones.

Config and exec commands, standard response fields (all optional)

`[errors]`	List of strings	If running this command produced any configuration errors, this field will be populated with a list of strings, each representing a discrete error.
`[warnings]`	List of strings	If running this command produced any warnings, this field will be populated with a list of strings, each representing a discrete warning.
`[messages]`	List of strings	Occasionally a non-show command will produce an informative message upon completion. If this happens, this field will be populated with a list of strings, each representing a discrete message.

Each following sub-chapter defines a cli command output, potentially followed by sub-sub-chapters that define the complex types used in the just described cli output (the title of those sub-sub-chapters are the complex type's name). The leftmost column is the attribute name, the middle column is its type, and the rightmost column is its description. Attributes can be of those basic types:

- string
- integer (any length)
- floating point number
- boolean (True or False)
- IPv4 address
- IPv6 address
- MAC address
- interface name
- enumeration (list of names)

and those complex types:

- list: ordered list
- table: unordered key/value list

Enumerations are represented like this: `enum: red | blue | green`. Tables are represented as `table(key)=type`, meaning the table is indexed by `key` and contains rows of type `type`.

2.1 clear security pki certificate rotation ROTATION_ID

`[errorMessage]`	*string*	Error message if not successful.
`rotationId`	*string*	Unique rotation ID.
`statusCode`	*enum: invalidRotationId | success*	Status code.

2.2 ping mpls bgp labeled-unicast (ip (IP_ADDR_PREFIX | (IP_PREFIX_PREFIX | (IP_PREFIX_PREFIX_ADDR IP_PREFIX_PREFIX_MASK))) | (ipv6 (IP6_ADDR_PREFIX6 | IP6_PREFIX_PREFIX6)) [...]

revision 2

ping	*MplsPingHdr* see 2.2.1	Ping request.

2.2.1 MplsPingHdr

[bgpLuModel] **	*BgpLuModel* see 2.2.1.1	MPLS BGP LU ping model. **Since: 4.30.0
[errInfo]	*list of ErrInfo* see 2.2.1.2	Errors.
[genericModel] **	*GenericModel* see 2.2.1.3	MPLS generic ping model. **Since: 4.30.0
interval	*integer*	Ping intervals.
protocol	*enum: segment-routing-ospf | nexthop-group-tunnel | rsvp | segment-routing | SrTePolicy | mldp | bgpLu | static | ldp | vpn | raw | generic | nexthop-group | pwLdp*	Protocol.
[pwLdpModel] **	*PwLdpModel* see 2.2.1.5	MPLS PW LDP ping model. **Since: 4.30.1
[rsvpModel] **	*RsvpModel* see 2.2.1.6	MPLS RSVP ping model. **Since: 4.30.0
[srModel] **	*SrModel* see 2.2.1.7	MPLS SR ping model. **Since: 4.30.0
statsModel	*list of MplsPingVia* see 2.2.1.4	MPLS ping statistics.
timeout	*integer*	Timeout for ping.
viaModel	*list of MplsPingVia* see 2.2.1.4	MPLS ping vias.
[vpnModel]	*VpnModel* see 2.2.1.8	MPLS VPN ping model.

2.2.1.1 BgpLuModel

prefix	*string*	Destination.

2.2.1.2 ErrInfo

error	*string*	Error encountered when cmd issued.

2.2.1.3 GenericModel

prefix	*string*	Destination.

2.2.1.4 MplsPingVia

[interface]	*string*	Interface name.
[labelStack]	*list of integer*	Label stack.
[lspId] **	*integer*	RSVP tunnel LSP ID. **Since: 4.30.0
[nextHopIp]	*string*	NextHop IP.
[nhgTunnelEntryIndex]	*integer*	Nexthop tunnel entry index.
[pingReply]	*MplsPingReply* see 2.2.1.4.1	MPLS reply packet details.
[pingStats]	*MplsPingStatisticsSummary* see 2.2.1.4.2	MPLS ping statistics.
resolved	*boolean*	Resolution status of tunnel/via.
[subTunnelId] **	*integer*	RSVP sub tunnel ID. **Since: 4.30.0

2.2.1.4.1 MplsPingReply

[errorSubTlvMap] **	*list of integer*	Error sub Tlv Map. **Since: 4.29.2
[errorTlvMap] **	*list of integer*	Error Tlv Map. **Since: 4.29.2
oneWayDelay **	*integer*	One way delay time for packet in microseconds. **Since: 4.29.1
replyHost	*string*	Reply host address.
retCode	*enum: noRetCode* | *seeDdmTlv* | *repRouterEgress* | *usIntfUnknown* | *noMappingForFec* | *labelSwitchedAtStackDep* | *labelSwitchedNoMplsFwd* | *malFormEchoRequest* | *reservedRetCode* | *labelMappingMismatch* | *protocolIntfMismatch* | *lblSwitchedFecChange* | *premTermination* | *noLabelEntryAtStackDep* | *dsMappingMismatch* | *tlvNotUnderstood*	Return code.
roundTripTime	*integer*	RoundTrip time for packet in milliseconds.
roundTripTimeUs **	*integer*	Round Trip time for packet in microseconds. **Since: 4.29.1
sequence	*integer*	Sequence number.

2.2.1.4.2 MplsPingStatisticsSummary

[mplsStatistics]	*list of MplsPingStatistics* see 2.2.1.4.2.1	MPLS ping statistics details.

packetLoss	*integer*	Percentage of packets lost.
packetsReceived	*integer*	Packets received.
packetsTransmitted	*integer*	Packets transmitted.
replyTime	*integer*	Reply timestamp time.

2.2.1.4.2.1 MplsPingStatistics

destination	*string*	Reply host address.
oneWayDelayAvg **	*integer*	Average one way delay in microseconds. **Since: 4.29.1
oneWayDelayMax **	*integer*	Maximum one way delay in microseconds. **Since: 4.29.1
oneWayDelayMin **	*integer*	Minimum one way delay in microseconds. **Since: 4.29.1
packetsReceived	*integer*	Packets received from destination.
roundTripTimeAvg	*integer*	Average round trip time in milliseconds.
roundTripTimeAvgUs **	*integer*	Average round trip time in microseconds. **Since: 4.29.1
roundTripTimeMax	*integer*	Maximum round trip time in milliseconds.
roundTripTimeMaxUs **	*integer*	Maximum round trip time in microseconds. **Since: 4.29.1
roundTripTimeMin	*integer*	Minimum round trip time in milliseconds.
roundTripTimeMinUs **	*integer*	Minimum round trip time in microseconds. **Since: 4.29.1

2.2.1.5 PwLdpModel

pwCCType	*enum: ra | cw | ttlExp | ccTypeUnset*	Pseudowire CC types.
pwId	*string*	Pseudowire ID.
pwLabel	*integer*	Pseudowire VC label.
pwLdpName	*string*	Pseudowire LDP name.
pwNeighbor	*string*	Neighbor IP address.

2.2.1.6 RsvpModel

[lspId]	*integer*	LSP Id.
[prefix]	*string*	Destination.
[session]	*string*	Rsvp session name or session Id.
[subTunnelId]	*integer*	Rsvp subtunnel Id.
[tunnel]	*string*	Rsvp tunnel name.

2.2.1.7 SrModel

[algorithm]	*string*	Algorithm.
[igpProtocolType] **	*string*	Segment Routing IGP Protocol. **Since: **currentRelease**
prefix	*string*	Destination.

2.2.1.8 VpnModel

prefix	*string*	Destination.
rd	*string*	Route Distinguisher.

2.3 ping mpls bgp vpn (vrf (VRF | default)) (ip PREFIX) | (ipv6 PREFIX6) [...]

revision 2

see 2.2

2.4 ping mpls generic ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...]

revision 2

see 2.2

2.5 ping mpls ldp ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...]

revision 2

see 2.2

2.6 ping mpls p2mp mldp ROOTADDR ((lsp-id OPQVAL) | (multicast ipv4 SOURCEADDR GROUPADDR)) [...]

revision 2

see 2.2

2.7 ping mpls pseudowire ((ldp PW_NAME) | (vpls ldp INSTANCE_NAME group GROUP_NAME NEIGHBOR_ADDRESS)) [...]

revision 2

see 2.2

2.8 ping mpls rsvp session ((id SESSION_ID) | (name SESSION_NAME)) [...]

revision 2

see 2.2

2.9 ping mpls rsvp tunnel TUNNEL_NAME [...]

revision 2

see 2.2

2.10 ping mpls segment-routing ((((ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (ipv6 IPV6_ADDRESS) | (isis ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (isis ipv6 IPV6_ADDRESS))) | (ospf ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))))) [...]

revision 2

see 2.2

2.11 reset management dmf controller sync

`controllerDigest`	*string*	SHA-256 digest of the config received from the controller.
`runningConfigDigest`	*string*	SHA-1 digest of the running-config of the switch.

2.12 security pki certificate generate signing-request rotation ssl profile PROFILE_NAME key generate ((rsa RSA_BIT_LENGTH) | (ecdsa EC_CURVE_NA)) [...]

`[csr]`	*string*	PEM encoded CSR.
`[errorMessage]`	*string*	Error message if not successful.
`[rotationId]`	*string*	Unique rotation ID.
`statusCode`	*enum: success | requestFailed*	Status code.

2.13 security pki certificate rotation commit ROTATION_ID

`[errorMessage]`	*string*	Error message if not successful.
`rotationId`	*string*	Unique rotation ID.

statusCode	*enum: invalidSslProfile* | *certNotYetValid* | *certExpired* | *certDoesNotExist* | *invalidRotationId* | *success* | *requestFailed*	Status code.

2.14 security pki certificate rotation commit ssl profile PROFILE_NAME [...]

[errorMessage]	*string*	Error message if not successful.
statusCode	*enum: invalidSslProfile* | *invalidKey* | *invalidCert* | *certNotYetValid* | *certExpired* | *success* | *certDoesNotMatchKey* | *requestFailed*	Status code.

2.15 security pki certificate rotation import ROTATION_ID

[errorMessage]	*string*	Error message if not successful.
rotationId	*string*	Unique rotation ID.
statusCode	*enum: invalidCert* | *certExpired* | *invalidRotationId* | *success* | *certDoesNotMatchKey* | *requestFailed*	Status code.

2.16 session peer-supervisor CMD

[output]	*string*	Output of the CLI command ran on the peer supervisor.

2.17 show ((ip bgp neighbors) | (ipv6 bgp peers)) ((received-routes) | (received-routes ((regex REGEX) | ((community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) })) | (access-list AS_PATH_LIST_NAME) | ((IPV4_ADDR | IPV6_ADDR)) | ((IPV4_ADDR_MASK | IPV6_ADDR_MASK))))) [...]

revision 3

vrfs	*table(string) = BgpRouteHeader* see 2.17.1	Per VRF BGP route summary.

2.17.1 BgpRouteHeader

asn	*string*	Autonomous System Number.
bgpRouteEntries	*table(string) = BgpRouteEntry* see 2.17.1.1	Dictionary of BGP route entries indexed by the route prefix. For SR-TE SAFI, the index is in the format: 'Endpoint|Color|Distinguisher'.
[currentSeqNum]	*integer*	Current route sequence number.
[lastProcessedSeqNum]	*integer*	Last route sequence number acknowledged.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF name.

2.17.1.1 BgpRouteEntry

address	*IP address*	Route address; For SR-TE SAFI, this is same as srTeEndpoint.
[bgpAdvertisedPeerGroups]	*table(string) = BgpRouteAdvertisedPeerListEntry* see 2.17.1.1.1	List of peers this route is advertised to indexed by peer-group name.
[bgpAdvertisedPeers]	*BgpRouteAdvertisedPeerListEntry* see 2.17.1.1.1	List of peers this route is advertised to.
bgpRoutePaths	*list of BgpRoutePath* see 2.17.1.1.2	List of BGP route ECMP paths.
[dpsAvtId] **	*integer*	Adaptive virtual topoligy ID of DPS policy. **Since: 4.28.2
[dpsAvtName] **	*string*	Adaptive virtual topoligy of DPS policy. **Since: 4.28.2
[dpsRouter] **	*IPv4 address*	Source router of DPS policy. **Since: 4.28.2
[dpsVni] **	*integer*	VNI of DPS policy. **Since: 4.28.2
[dpsVrf] **	*string*	VRF of DPS policy. **Since: 4.28.2
maskLength	*integer*	Route mask length.

[notUcmpEligibleReason]	*enum: noLinkBWOnAllPaths | maxUcmpExceeded*	Used when ECMP could be converted to UCMP. NoLinkBWOnAllPaths - All contributing paths do not have link bandwidth available maxUcmpExceeded - Max UCMP configuration is less than or equal to the number of nexthops preventing UCMP application effectively.
[srTeColor]	*integer*	Color of SR TE policy.
[srTeDistinguisher]	*integer*	Distinguisher of SR TE policy.
[srTeEndpoint]	*IP address*	Endpoint address of SR TE policy.
[totalAdvertisedPeers]	*integer*	Number of peers this route is advertised to.
[totalPaths]	*integer*	Total number of paths for this route.

2.17.1.1.1 BgpRouteAdvertisedPeerListEntry

peerList	*list of string*	List of peers the route is advertised to.

2.17.1.1.2 BgpRoutePath

[aigpDistance] **	*integer*	Accumulated IGP Distance for the route. **Since: 4.24.1
[aigpMetric] **	*integer*	Accumulated IGP metric for the route. **Since: 4.24.1
asPathEntry	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[nextHop]	*string*	Route next hop address.
[orrWinners] **	*list of string*	List of ORR positions for which this path is best. **Since: 4.27.0
[peerEntry] **	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route. **Since: 4.21.3

reasonNotBestpath	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*BgpRouteDetailEntry* see 2.17.1.1.2.2	Route details.
routeType	*BgpRouteTypeEntry* see 2.17.1.1.2.36	Route type.
[tag] **	*integer*	Tag for this route. **Since: 4.26.0
[timestamp]	*integer*	UTC seconds since epoch when the route was received. Only returned with 'show ip bgp detail'.
[weight]	*integer*	Weight for the route.

2.17.1.1.2.1 BgpRouteASPathEntry

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
asPathType	*enum: Invalid* | *External* | *Confed-External* | *Internal* | *Local*	AS path type: Internal - originated by I-BGP External - originated by E-BGP Confed-External - originated by a E-BGP confederation Local - originated locally Invalid - AS path is invalid.

2.17.1.1.2.2 BgpRouteDetailEntry

[aggregator]	*string*	Aggregator of the route.

[aigpDistance] **	*integer*	Accumulated IGP Distance for the route. **Since: 4.24.1
[aigpMetric] **	*integer*	Accumulated IGP metric for the route. **Since: 4.24.1
[bgpContributors]	*list of BgpRouteContributor* see 2.17.1.1.2.22	List of contributors to this aggregated route.
[bgpLsLinkAttributes] **	*BgpLsLinkAttributes* see 2.17.1.1.2.3	Attributes a BGP Link State link. **Since: 4.24.2
[bgpLsNodeAttributes] **	*BgpLsNodeAttributes* see 2.17.1.1.2.8	Attributes of a BGP Link State node. **Since: 4.24.2
[bgpLsPrefixAttributes] **	*BgpLsPrefixAttributes* see 2.17.1.1.2.18	Attributes of a BGP Link State prefix. **Since: 4.24.2
[clusterList]	*string*	Cluster list for the route.
communityList	*list of string*	List of communities for this route.
[domainPath] **	*list of string*	Domain path attribute. **Since: 4.23.0
extCommunityList	*list of string*	List of extended communities for this route.
[extCommunityListRaw] **	*list of integer*	List of raw extended communities for this route. **Since: 4.25.2
[fieldSetName] **	*string*	Field Set assigned to the route. **Since: 4.27.1
[igpMetric]	*integer*	IGP metric.
[isLeaked] **	*boolean*	Route leaked from a different VRF. **Since: 4.23.2
[labelStack]	*list of integer*	MPLS label stack information.
largeCommunityList **	*list of string*	List of large communities for this route. **Since: 4.21.3
[localLabel]	*integer*	Local MPLS label.
[nextHopVrfName] **	*string*	VRF Name of the nexthop for the route. **Since: 4.21.3
origin	*enum: Igp | Egp | Incomplete*	Route origin.
[originator]	*string*	Router ID of the originator of this route.
[pendingTimeToAdv]	*floating point number*	Timestamp of route advertisement.
[pmsiTunnel] **	*PmsiTunnel* see 2.17.1.1.2.31	P-Multicast Service Interface Tunnel Attribute. **Since: 4.23.2
[recmpEligible] **	*boolean*	The route is RECMP eligible. **Since: 4.29.1
recvdFromRRClient	*boolean*	Route received from route reflector client.
[redistributionProtocol]	*enum: Connected | Ospf3 | Static | Bgp | unknown | Is-Is | Rip | Dynamic | Ldp | Is-Is-Sr | BGP-Aggregate | Eos-Sdk | Ospf*	Protocol from which the route was redistributed into BGP.
[remoteLabel]	*EncapLabel* see 2.17.1.1.2.28	Remote label.
[rfdFlaps] **	*integer*	Number of path flaps. **Since: 4.29.2

[rfdPenalty] **	*integer*	Route flap damping penalty. **Since: 4.29.2
[rxPathId]	*integer*	Received path ID of this route.
[rxSafi]	*string*	Received SAFI of this route.
[seqNumber]	*integer*	Route sequence number.
[srLabelIndex]	*integer*	Prefix SID Label Index.
[srTeTunnelEncap]	*BgpSrTePolicyTunnelEncap* see 2.17.1.1.2.23	Bgp Segment Routing Policy Tunnel Encapsulation Attribute.
tunnelRibEligible	*boolean*	Route eligible to be installed in Tunnel RIB.
[txPathId]	*integer*	Advertised path ID for this route.

2.17.1.1.2.3 BgpLsLinkAttributes

[administrativeGroup]	*integer*	Administrative Group(color) of link.
[dpsLinkAttributes] **	*DpsLinkAttributes* see 2.17.1.1.2.7	DPS Link Attributes Information. **Since: 4.28.2
igpMetric	*integer*	IGP metric of link.
[localIpv4RouterId]	*IPv4 address*	IPv4 Router ID of local node.
[localIpv6RouterId]	*IPv6 address*	IPv6 Router ID of local node.
[maxLinkBw]	*integer*	Maximum link bandwidth (bps).
[maxReservableLinkBw]	*integer*	Maximum bandwidth (bps) that can be reserved on link.
[remoteIpv4RouterId]	*IPv4 address*	IPv4 Router ID of remote node.
[remoteIpv6RouterId]	*IPv6 address*	IPv6 Router ID of remote node.
[srAdjacencySids]	*list of BgpLsSrAdjacencySid* see 2.17.1.1.2.4	List of SR Adjacency SIDs.
[srlgIds]	*list of integer*	Link SRLG values.
[teDefaultMetric]	*integer*	Traffic engineering metric of link.
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Link Path Attribute TLVs. **Since: 4.30.1
[unreservedBw]	*table(string) = integer*	Unreserved bandwith(bps) keyed by priority.

2.17.1.1.2.4 BgpLsSrAdjacencySid

flags	*BgpLsSrAdjSidFlags* see 2.17.1.1.2.5	SR Adjacency SID Flags.
sid	*integer*	SR Adjacency Index/Label SID.
weight	*integer*	Weight of the Adj-SID for load balancing.

2.17.1.1.2.5 BgpLsSrAdjSidFlags

adjacencySet	*boolean*	SID refers to set of adjacencies and may also be assigned to other adjacencies.
backup	*boolean*	SID is eligible for protection such as using IPFRR or MPLS-FRR.
ipv6	*boolean*	SID is used for forwarding IPv6 traffic.
local	*boolean*	Value/index carried by SID has local significance.

persistent	*boolean*	SID is persistent across router restart/interface flap.
value	*boolean*	SID carries a value instead of an index.

2.17.1.1.2.6 BgpLsUnknownTlv

tlvLen	*integer*	Unknown TLV Length.
tlvType	*integer*	Unknown TLV Type.

2.17.1.1.2.7 DpsLinkAttributes

[pathJitter]	*integer*	DPS Path Jitter (microseconds).
[pathLatency]	*integer*	DPS Path Latency (microseconds).
[pathLossRate]	*integer*	DPS Path Loss Rate (percentage).
[pathMtu]	*integer*	DPS Path MTU.
[segmentLabel]	*integer*	DPS Segment Label.

2.17.1.1.2.8 BgpLsNodeAttributes

[dpsNodeAttributes] **	*DpsNodeAttributes* see 2.17.1.1.2.10	DPS Node Attributes Information. **Since: 4.28.2
flags	*BgpLsNodeFlags* see 2.17.1.1.2.9	Node flags.
[ipv4RouterId]	*IPv4 address*	IPv4 router ID of node.
[ipv6RouterId]	*IPv6 address*	IPv6 router ID of node.
isisAreaIds	*list of string*	List of IS-IS area identifiers.
nodeMsds **	*list of MsdModel* see 2.17.1.1.2.12	List of Node MSDs. **Since: 4.28.1
[nodeName]	*string*	Symbolic name of node.
[srCapabilities]	*SrCapabilityModel* see 2.17.1.1.2.13	SR Capabilities.
[srlb]	*SrlbModel* see 2.17.1.1.2.16	SR Local Block Information.
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Node Path Attribute TLVs. **Since: 4.30.1

2.17.1.1.2.9 BgpLsNodeFlags

isisAttached	*boolean*	IS-IS node is attached to L2 node in a different area.
isisOverload	*boolean*	IS-IS node is overloaded.

2.17.1.1.2.10 DpsNodeAttributes

[avtIds]	*list of DpsAvtIdsModel* see 2.17.1.1.2.11	DPS AVT ID.
[regionId]	*integer*	DPS Region ID.

role	*enum: zone transit | edge | unknown | pathfinder | region transit*	DPS Role.
[siteId]	*integer*	DPS Site ID.
[zoneId] **	*integer*	DPS Zone ID. **Since: 4.29.0

2.17.1.1.2.11 DpsAvtIdsModel

dpsAvtId	*integer*	DPS AVT ID.
dpsAvtVni	*integer*	DPS AVT VNI.

2.17.1.1.2.12 MsdModel

msdType	*integer*	MSD type.
msdValue	*integer*	MSD value.

2.17.1.1.2.13 SrCapabilityModel

flags	*SrCapabilityFlags* see 2.17.1.1.2.14	SR Capability Flags.
srCapabilitySrgb	*list of SrCapabilitySrgb* see 2.17.1.1.2.15	Segment Routing Global Blocks.

2.17.1.1.2.14 SrCapabilityFlags

mplsV4	*boolean*	SR capability MPLS IPv4 Flag.
mplsV6	*boolean*	SR capability MPLS IPv6 Flag.

2.17.1.1.2.15 SrCapabilitySrgb

srgbBase	*integer*	SRGB base.
srgbRange	*integer*	SRGB range.

2.17.1.1.2.16 SrlbModel

srlbRanges	*list of SrlbRange* see 2.17.1.1.2.17	Segment Routing Local Blocks.

2.17.1.1.2.17 SrlbRange

srlbBase	*integer*	SRLB base.
srlbRange	*integer*	SRLB range.

2.17.1.1.2.18 BgpLsPrefixAttributes

flags	*BgpLsPrefixFlags* see 2.17.1.1.2.19	Prefix flags.
[metric]	*integer*	Metric of prefix as known in the IGP topology.
[srPrefixSid]	*BgpLsSrPrefixSid* see 2.17.1.1.2.20	SR Prefix SID.
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Prefix Path Attribute TLVs. **Since: 4.30.1

2.17.1.1.2.19 BgpLsPrefixFlags

isisDown	*boolean*	A prefix has been leaked from a higher level to a lower level.

2.17.1.1.2.20 BgpLsSrPrefixSid

algorithm	*integer*	Algorithm identifier to compute reachability of prefix.
flags	*BgpLsSrPrefixFlags* see 2.17.1.1.2.21	SR Prefix SID Flags.
sid	*integer*	SR Prefix Index/Label SID.

2.17.1.1.2.21 BgpLsSrPrefixFlags

explicitNull	*boolean*	Upstream neighbor must replace with an Explicit NULL value SID before sending packet.
local	*boolean*	Value/index carried by SID has local significance.
noPhp	*boolean*	Penultimate hop must not pop SID before sending packet to advertising node.
nodeSid	*boolean*	SID refers to the router identified by the prefix, typically attached to router loopback address.
readvertised	*boolean*	Corresponding prefix has been re-advertised from another level or redistribution.
value	*boolean*	SID carries a value instead of an index.

2.17.1.1.2.22 BgpRouteContributor

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
communities	*list of string*	List of communities for the contributor.
contributorAddr	*IP address*	Contributor address.
contributorMaskLen	*integer*	Contributor mask length.
extCommunities	*list of string*	List of extended communities for the contributor.

[extCommunityListRaw] **	*list of integer*	List of raw extended communities for the contributor. **Since: 4.26.0
largeCommunities **	*list of string*	List of large communities for the contributor. **Since: 4.21.3
[nlri] **	*IP address prefix*	Contributor prefix. **Since: 4.26.0
origin	*enum: Igp | Egp | Incomplete*	Contributor Route origin.
[redistProtocol]	*enum: Connected | Ospf3 | Static | Bgp | unknown | Is-Is | Rip | Dynamic | Ldp | Is-Is-Sr | BGP-Aggregate | Ospf*	Protocol from which the contributor was learned.

2.17.1.1.2.23 BgpSrTePolicyTunnelEncap

[bindingSid]	*BgpSrTePolicyBindingSid* see 2.17.1.1.2.24	Binding Segment Identifier.
[enlp] **	*BgpSrTePolicyEnlp* see 2.17.1.1.2.25	Explicit null label policy. **Since: 4.23.0
[preference]	*integer*	Preference of the SR Policy.
segmentLists	*list of BgpSrTePolicySegmentList* see 2.17.1.1.2.26	Segment lists.

2.17.1.1.2.24 BgpSrTePolicyBindingSid

bindingSidType	*enum: mpls | ipv6*	Binding Segment Identifier Type.
[mplsLabelSid]	*integer*	Binding Segment Identifier Value in the form of MPLS Label.

2.17.1.1.2.25 BgpSrTePolicyEnlp

enlpType	*enum: unknown | ipv4 | ipv6 | ipv4AndIpv6 | none*	Explicit null label policy type.
value	*integer*	Explicit null label policy value.

2.17.1.1.2.26 BgpSrTePolicySegmentList

[jitter] **	*floating point number*	Jitter of the path in msecs. **Since: 4.28.2
[latency] **	*floating point number*	Latency of the path in msecs. **Since: 4.28.2
[lossRate] **	*floating point number*	Loss rate of the path in percentage. **Since: 4.28.2
[mtu] **	*integer*	Maximum transmission unit of the path in bytes. **Since: 4.28.2
[priority] **	*integer*	Priority of the path. **Since: 4.29.0

segments	*list of BgpSrTePolicySegment* see 2.17.1.1.2.27	Segments that make up the segment list.
[weight]	*integer*	Weight of the segment list.

2.17.1.1.2.27 BgpSrTePolicySegment

[mplsLabelSid]	*integer*	Segment Identifier in the form of MPLS Label.

2.17.1.1.2.28 EncapLabel

encapType	*enum: SRv6 | VXLAN | MPLS*	Encapsulation type.
[srv6Sid] **	*Srv6Sid* see 2.17.1.1.2.29	SRv6 SID. **Since: 4.28.2
value	*string*	Label value.

2.17.1.1.2.29 Srv6Sid

behavior	*enum: endDt6 | unknown | end | endX | endDx6 | endDt4 | endDx4*	SID behavior.
[flavors]	*Srv6SidFlavorFlags* see 2.17.1.1.2.30	SID flavors.
sid	*IP address prefix*	SID value (IPv6 prefix).

2.17.1.1.2.30 Srv6SidFlavorFlags

[nextCsid]	*boolean*	NEXT-CSID flavor.
[nextOnlyCsid] **	*boolean*	NEXT-ONLY-CSID flavor. **Since: 4.30.0
[psp]	*boolean*	Penultimate Segment Pop (PSP) of the SRH flavor.
[usd]	*boolean*	Ultimate Segment Decapsulation (USD) flavor.
[usp]	*boolean*	Ultimate Segment Pop (USP) of the SRH flavor.

2.17.1.1.2.31 PmsiTunnel

leafInformationRequired	*boolean*	Leaf information required.
mplsLabel	*integer*	MPLS label.
[tunnelId]	*PmsiTunnelId* see 2.17.1.1.2.32	Tunnel information.

tunnelType	*enum: assistedReplication* | *rsvpTeP2mpLsp* | *bier* | *pimSmTree* | *unknown* | *pimSsmTree* | *ingressReplication* | *noInfo* | *mLdpP2mpLsp* | *mLdpMp2mpLsp* | *bidirPimTree* | *transportTunnel* | *wildcardTransport*	Tunnel technology used to establish PMSI tunnel.

2.17.1.1.2.32 PmsiTunnelId

[ingressReplication]	*IP address*	Replication address.
[mldpP2mp] **	*PmsiMldpTunnelId* see 2.17.1.1.2.33	MLDP P2MP tree information. **Since: 4.27.0
[pim] **	*PmsiPimTunnelId* see 2.17.1.1.2.34	PIM tree information. **Since: 4.24.2

2.17.1.1.2.33 PmsiMldpTunnelId

opaqueValue	*string*	FEC opaque value.
rootAddress	*IP address*	FEC root address.

2.17.1.1.2.34 PmsiPimTunnelId

group	*IP address*	Destination address used in tunnel encapsulation.
source	*IP address*	Source address used in tunnel encapsulation.

2.17.1.1.2.35 BgpRoutePeerEntry

peerAddr	*string*	Peer address for the route.
peerRouterId	*IPv4 address*	Peer router Identity.

2.17.1.1.2.36 BgpRouteTypeEntry

active	*boolean*	Route is active.
atomicAggregator	*boolean*	Route is an atomic-aggregate.
backup	*boolean*	Route is backup.
ecmp	*boolean*	Route is an ECMP route.
ecmpContributor	*boolean*	Route contributes to ECMP.
ecmpHead	*boolean*	Route is the ECMP head.
[localAgg]	*boolean*	Route is locally aggregated.
luRoute	*boolean*	Route is an LU route.
[notInstalledReason]	*enum: labeledRoutePresent* | *routeInstallDenied* | *routeBestInactive*	Reason for route not being installed.

[origin]	*enum: Igp | Egp | Incomplete*	Route origin.
[originValidity] **	*enum: valid | invalid | notFound | notValidated*	Result of validating origin AS. **Since: 4.26.0
queued	*boolean*	Route is queued for advertisement.
[rfdSuppressed] **	*boolean*	Path is suppressed as it exceeded route flap damping suppression threshold. **Since: 4.29.2
[sixPeRoute] **	*boolean*	Route is an IPv6 provider edge (6PE) route. **Since: 4.22.1
stale	*boolean*	Route is stale.
suppressed	*boolean*	Route is suppressed by a covering aggregate.
ucmp	*boolean*	Route is an UCMP route.
[unknownMetric] **	*boolean*	Route is pending resolution. **Since: 4.24.1
valid	*boolean*	Route is valid.
[waitForConvergence]	*boolean*	Route is pending BGP convergence.

2.18 show ((ip bgp neighbors IP_PEER) | (ipv6 bgp peers (IP6_PEER | IP6_LL_PEER))) advertised-routes queued-withdrawals [...]

vrfs	*table(string) = BgpQueuedWithdrawals* see 2.18.1	Per VRF BGP queued withdrawals.

2.18.1 BgpQueuedWithdrawals

bgpQueuedWithdrawals	*list of BgpQueuedWithdrawal* see 2.18.1.1	List of paths pending withdrawal due to out-delay.
localAsn	*string*	Local Autonomous System Number.
localRouterId	*IPv4 address*	Local BGP Router Identity.
peerAddress	*string*	BGP peer address.
vrf	*string*	VRF name.

2.18.1.1 BgpQueuedWithdrawal

address	*IP address*	Address of route being withdrawn.
[localLabel] **	*integer*	Local MPLS label of route being withdrawn. **Since: 4.23.1
maskLength	*integer*	Mask length of route being withdrawn.
[pathId]	*integer*	Path ID of route being withdrawn.
rxAfiSafi **	*string*	Received AFI SAFI of route being withdrawn. **Since: 4.23.1
secondsRemaining	*integer*	Seconds remaining until route is withdrawn.

2.19 show ((ip bgp neighbors PEER) | (ipv6 bgp peers ((IP6_PEER | IP6_LL_PEER)))) (((routes | advertised-routes)) | ((routes | advertised-routes) ((regex REGEX) | ((community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) })) | (access-list AS_PATH_LIST_NAME) | ((IPV4_ADDR | IPV6_ADDR)) | ((IPV4_ADDR_MASK | IPV6_ADDR_MASK))))) [...]

revision 3

see 2.17

2.20 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) IP1 IP2 interface [...]

group	*IP address*	Group address of multicast route.
interfaces	*table(Interface name) = MrouteSmInterface* see 2.20.1	Map of interfaces for multicast route.
source	*IP address*	Source address of multicast route.

2.20.1 MrouteSmInterface

[address]	*IP address*	Interface IP address.
assertState	*enum: iAmAssertLoser | assertNoInfo | iAmAssertWinner*	Assert state.
[details]	*MrouteSmInterfaceDetails* see 2.20.1.1	Detailed mroute interface information.
downstreamJoinState	*enum: noInfo | prunePending | joined*	Downstream join state.
forward	*boolean*	Member of outgoing interface set.
iAmDr	*boolean*	This Pim router is the designated router on this interface.
[lastJoinReceived]	*floating point number*	UTC time at which the last downstream join was received.
[lastRptPruneReceived]	*floating point number*	UTC time at which the last downstream RPT prune was received.
markedForDeletion	*boolean*	If true, this interface has no (S,G,I)/(*,G,I) state and is scheduled to be removed from the route's interface list.
rptState	*enum: rptPruneTemp | rptPrunePendingTemp | rptPruned | rptNoInfo | rptPrunePending*	Downstream RPT state.

2.20.1.1 MrouteSmInterfaceDetails

[assertWinnerMetric]	*AssertMetric* see 2.20.1.1.1	Assert winner metric.
[evpnInfo] **	*EvpnInfo* see 2.20.1.1.2	EVPN Information. **Since: 4.25.0
[lastAssertAction]	*integer*	Last assert action.
[sgState]	*SgMrouteInterface* see 2.20.1.1.3	Interface state specific to (S,G) routes.
[wcState]	*WcMrouteInterface* see 2.20.1.1.4	Interface state specific to (*,G) routes.

2.20.1.1.1 AssertMetric

address	*IP address*	Address of router that generated the assert.
metric	*integer*	Route metric to the source/RP.
preference	*integer*	Route preference to the source/RP.

rptBit	*boolean*	Assert metric is for the RP tree.

2.20.1.1.2 EvpnInfo

[pegDr]	*boolean*	This interface is PEG DR.
[sbdVlan]	*boolean*	This is SBD VLAN interface.
[vni]	*integer*	VNI of the interface.
[vrf]	*string*	VRF the interface belongs to.

2.20.1.1.3 SgMrouteInterface

assertTrackingDesiredSg	*boolean*	Assert tracking desired (S,G,I).
couldAssertSg	*boolean*	Could assert (S,G,I).
excludeSg	*boolean*	Member of pim exclude (S,G).
immediateOlistSg	*boolean*	Member of immediate olist (S,G).
includeSg	*boolean*	Member of pim include (S,G).
inheritedOlistSg	*boolean*	Member of inherited olist (S,G).
inheritedOlistSgRpt	*boolean*	Member of inherited olist (S,G,Rpt).
joinsSg	*boolean*	Member of joins (S,G).
localReceiverExcludeSg	*boolean*	Local receiver exclude (S,G,I).
localReceiverIncludeSg	*boolean*	Local receiver include (S,G,I).
lostAssertSg	*boolean*	Member of lost assert (S,G).
lostAssertSgRpt	*boolean*	Member of lost assert (S,G,Rpt).
[myAssertMetricSg]	*AssertMetric* see 2.20.1.1.1	My assert metric (S,G,I).
prunesSgRpt	*boolean*	Member of prunes (S,G,Rpt).
[sptAssertMetric]	*AssertMetric* see 2.20.1.1.1	SPT assert metric (S,I).

2.20.1.1.4 WcMrouteInterface

assertTrackingDesiredWc	*boolean*	Assert tracking desired (*,G,I).
couldAssertWc	*boolean*	Could assert (*,G,I).
immediateOlistWc	*boolean*	Member of immediate olist (*,G).
immediateOlistWcRp	*boolean*	Member of immediate olist (*,*,G).
includeWc	*boolean*	Member of pim include (*,G).
joinsWc	*boolean*	Member of joins (*,G).
joinsWcRp	*boolean*	Member of joins (*,*,G).
localReceiverIncludeWc	*boolean*	Local receiver include (*,G,I).
lostAssertWc	*boolean*	Member of lost assert (*,G).
[myAssertMetricWc]	*AssertMetric* see 2.20.1.1.1	My assert metric (*,G,I).
[rptAssertMetric]	*AssertMetric* see 2.20.1.1.1	RPT assert metric (G,I).

2.21 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) [...]

groups	*table(IPv4 or IPv6 address) = GroupSm* see 2.21.1	Map of sparse-mode Pim multicast groups.

2.21.1 GroupSm

groupSources	*table(IPv4 or IPv6 address) = MrouteSm* see 2.21.1.1	Map of multicast routes for this group.

2.21.1.1 MrouteSm

creationTime	*floating point number*	UTC time at which the route was created.
[details]	*MrouteDetails* see 2.21.1.1.1	Detailed information for this multicast route.
[nonDrOifList] **	*list of Interface name*	List of outgoing interfaces with egress drop rule in effect on non-DR. **Since: 4.22.1
oifList	*list of Interface name*	Outgoing interface list.
[registerInOifList]	*boolean*	If true, the register tunnel is part of the outgoing interface list.
routeFlags	*string*	Route flags.
[rp]	*IP address*	RP(G): The Rendezvous Point for this group.
[rpf]	*RpfSm* see 2.21.1.1.2	Reverse path forwarding information.
[rpfFrr] **	*RpfSm* see 2.21.1.1.2	Reverse path forwarding information for MoFrr. **Since: 4.20.0
[rpfFrrInterface] **	*interface name*	Reverse path forwarding interface for MoFRR. **Since: 4.20.0
[rpfInterface]	*interface name*	Reverse path forwarding interface, packets for this mroute must arrive on this interface to be accepted/forwarded. If this value is set to 'Register', the RPF interface is the register tunnel.
sourceAddress	*IP address*	Source address.

2.21.1.1.1 MrouteDetails

[evpnInfo] **	*EvpnInfo* see 2.20.1.1.2	EVPN Information. **Since: 4.25.0
[fastFailovers] **	*list of string*	This field is not used. **Since: 4.22.0
oifPrimes	*table(string) = OifPrime* see 2.21.1.1.1.1	Map of interfaces not in OIF set.
[sgState]	*SgMroute* see 2.21.1.1.1.2	State specific to (S,G) routes.

upstreamJoinState	*enum: upNotJoined | upJoined*	Upstream joined state.
upstreamRptState	*enum: upRptPruned | upRptNotPruned | upRptNotJoined*	Upstream RPT joined state.
[wcState]	*WcMroute* see 2.21.1.1.1.3	State specific to (*,G) routes.

2.21.1.1.1.1 OifPrime

assertLoser	*boolean*	Assert loser on this interface.
incomingInterface	*boolean*	The incoming interface for the route.
interface	*interface name*	Name of this interface.
locallyExcluded	*boolean*	Locally excluded.
notDr	*boolean*	Not a designated router on this interface.
rptPruned	*boolean*	Pruned from the RP tree.

2.21.1.1.1.2 SgMroute

couldRegisterSg	*boolean*	CouldRegister(S,G): If true, we may send register messages for this (S,G).
joinDesiredSg	*boolean*	Join Desired (S,G), if true we should send joins for this source if a valid upstream neighbor exists.
nullImmediateOlistSg	*boolean*	No interfaces in immediate olist (S,G).
nullInheritedOlistSg	*boolean*	No interfaces in immediate olist (S,G).
pruneDesiredSgRpt	*boolean*	Prune Desired (S,G,Rpt), if true, we should prune ourselves from the RP tree for this source. This is to avoid receiving traffic both from the RP tree and source tree.
registerStopSg	*boolean*	If true, we should send a register stop for this (S,G) in response to a register message.

2.21.1.1.1.3 WcMroute

joinDesiredWc	*boolean*	Join Desired (*,G).
joinDesiredWcRp	*boolean*	Join Desired (*,*,G).
nullImmediateOlistWc	*boolean*	No interfaces in immediate olist (*,G).
nullImmediateOlistWcRp	*boolean*	No interfaces in immediate olist (*,*,G).
rptJoinDesiredWc	*boolean*	RPT Join Desired (*,G).

2.21.1.1.2 RpfSm

rpfAttached	*boolean*	If true, the source/RP is directly connected to this router.
[rpfEvpnTenantDomain] **	*boolean*	If true, the source/RP is inside EVPN Tenant Domain. **Since: 4.23.1
rpfMetric	*integer*	RPF route metric.

[rpfMvpn] **	*boolean*	If true, the source/RP is inside MVPN Domain. **Since: 4.27.1
[rpfNeighbor]	*IP address*	RPF neighbor.
rpfPreference	*integer*	RPF route preference.
rpfPrefix	*IP address prefix*	RPF route prefix.
rpfRib	*enum: M | none | U*	RPF routing table.

2.22 show ((ip mroute sparse-mode) | (pim ipv4 sparse-mode route)) count [...]

anycastRpDiscovered	*integer*	Number of routes discovered on the RP from from an anycast RP via a register message.
couldRegister	*integer*	Number of (S,G)s for which CouldRegister(S,G) is True.
drDiscovered	*integer*	Number of routes discovered on the RP from a DR via aa register message.
groups	*integer*	Number of groups.
mbrDiscovered	*integer*	Number of (S,G)s discovered via MBR.
msdpDiscovered	*integer*	Number of routes discovered on the RP via an MSDP source advertisement.
registerOif	*integer*	Number of (S,G)s with the register tunnel in the OIF set.
registerStop	*integer*	Number of (S,G)s for which the RP should generate register stops on receipt of a register/null register.
sgRoutes	*integer*	Number of (S,G) routes.
sources	*integer*	Number of sources.
switching	*integer*	Number of (S,G)s currently in the midst of switching from the RP tree to the SP tree.
toBeDeleted	*integer*	Number of routes currently scheduled for deletion.
unresolved	*integer*	Number of routes for which RPF information has not beendetermined.
wcRoutes	*integer*	Number of (*,G) routes.

2.23 show ((ip pim) | (pim ipv4)) sparse-mode rp [...]

crpSet	*table(IPv4 or IPv6 address prefix) = PimRpCandidateSet* see 2.23.1	Map of Candidate RP Set belonging to this group prefix.
detail	*boolean*	ShowDetails.

2.23.1 PimRpCandidateSet

crp	*table(IPv4 or IPv6 address) = PimRpCandidate* see 2.23.1.1	Map of candidate RP information for this RP.
[mode]	*enum: Bidir | Sparse/Bidir | None | Sparse*	PIM mode.
prefix	*IP address prefix*	Group Prefix.

2.23.1.1 PimRpCandidate

creationTime	*floating point number*	UTC time at which the entry was created.
[expires]	*floating point number*	Number of seconds after which the entry will expire.
hashMaskLen	*integer*	Hash Mask Length.
[hashMaskValue]	*integer*	Hash Mask Length.
[holdTime]	*integer*	Number of seconds for which this entry is set to be held.
[mode]	*enum: Bidir | Sparse/Bidir | None | Sparse*	PIM mode.
[override]	*boolean*	Entry is set to override BSR provided RP.
priority	*integer*	Rp Priority.
rpAddress	*IP address*	IP address of RP.
rpType	*string*	Type of RP candidate.

2.24 show ((ip pim sparse-mode rp-hash) | (pim ipv4 sparse-mode rp hash)) GROUP [...]

[crp]	*table(IPv4 or IPv6 address) = PimRpCandidate* see 2.23.1.1	Map of candidate RP info related to the RP Hash.
detail	*boolean*	Show details.
[mode]	*enum: modePimNone | modePimSm | modePimBidir*	PIM mode.
prefix	*IP address prefix*	Group Prefix.
[rp]	*IP address*	Rendezvous Point Router's IP address.
[rpHashAlgorithm] **	*enum: rpHashAlgorithmModulo | rpHashAlgorithmDefault*	Rp Hash Algorithm. **Since: 4.17.0
[rpHashMaskLen]	*integer*	Hash Mask Length.
[rpPriority]	*integer*	RP Priority.

2.25 show ((ip pim sparse-mode upstream joins) | (pim ipv4 sparse-mode join)) [...]

neighbors	*table(IPv4 or IPv6 address) = JoinPruneNeighbor* see 2.25.1	Map of upstream neighbors to which periodic join/prunes are being sent.

2.25.1 JoinPruneNeighbor

groups	*table(IPv4 or IPv6 address) = JoinPruneGroup* see 2.25.1.1	Map of groups for which periodic joins/prunes are being sent to this neighbor.
interface	*interface name*	Interface via which joins/prunes are being sent to this neighbor.
interfaceAddress	*IP address*	The join/prune interface's address.

2.25.1.1 JoinPruneGroup

joinedSources	*table(IPv4 or IPv6 address prefix) = JoinPruneSource* see 2.25.1.1.1	Map of sources for this group for which periodic joins are being sent.
[mode]	*enum: Bidir | Sparse/Bidir | None | Sparse*	PIM mode.
prunedSources	*table(IPv4 or IPv6 address prefix) = JoinPruneSource* see 2.25.1.1.1	Map of sources for this group for which periodic prunes are being sent.

2.25.1.1.1 JoinPruneSource

joinPruneFlagList	*list of JoinPruneFlag* see 2.25.1.1.1.1	Join/prune flags.
joinSuppressed	*boolean*	The upstream join for this source is currently suppressed.

2.25.1.1.1.1 JoinPruneFlag

flag	*enum: RPT | WC | SPT*	Type of join/prune, can be WC => (*,G), RPT => (S,G,Rpt) or SPT => (S,G).

2.26 show ((mac address-table) | mac-address-table) [...]

revision 2

disabledMacLearningVlans	*list of integer*	VLANs with disabled MAC learning.
[multicastTable]	*MulticastMacAddressTable* see 2.26.1	Multicast MAC address table.
unicastTable	*UnicastMacAddressTable* see 2.26.2	Unicast MAC address table.

2.26.1 MulticastMacAddressTable

tableEntries	*list of MulticastTableEntry* see 2.26.1.1	List of all of the Multicast entries.

2.26.1.1 MulticastTableEntry

entryType	*static*	Type.
interfaces	*list of Interface name*	Interfaces of the MAC Address.
macAddress	*MAC address*	MAC Address.
vlanId	*integer*	VLAN Id of MAC address.

2.26.2 UnicastMacAddressTable

tableEntries	*list of UnicastTableEntry* see 2.26.2.1	List of all of the Unicast entries.

2.26.2.1 UnicastTableEntry

entryType	*enum: dynamic | static*	MAC Address creation type.
interface	*interface name*	Interface of the MAC Address.
[interfaceFunction] **	*string*	MAC address interface function. **Since: 4.26.2
[lastMove]	*floating point number*	Timestamp of last move.
macAddress	*MAC address*	MAC Address.
[moves]	*integer*	Number of moves.
vlanId	*integer*	VLAN Id of MAC address.

2.27 show ((mac address-table) | mac-address-table) aging-time

hostAgingTime	*floating point number*	Host Aging Time.

2.28 show ((mac address-table) | mac-address-table) count [...]

vlanCounts	*table(numeric string) = VlanCount* see 2.28.1	Mapping between a VLAN Id and it's counters.

2.28.1 VlanCount

dynamic	*integer*	Number of dynamic address MAC entries.
multicast	*integer*	Number of multicast address MAC entries.
unicast	*integer*	Number of unicast address MAC entries.

2.29 show ((mac address-table) | mac-address-table) dynamic local limit [...]

revision 2

limitedVlans	*table(numeric string) = LimitedVlanEntry* see 2.29.1	Limited VLANs, keyed by their ID.

2.29.1 LimitedVlanEntry

appliedLimit	*integer*	Limit programmed in the hardware.
currentAddresses	*integer*	Current number of locally learned MACs.
limit	*integer*	Maximum number of locally learned MACs.

2.30 show ((mac address-table) | mac-address-table) mlag-peer [...]

see 2.26.2

2.31 show ((mac address-table) | mac-address-table) mlag-peer diff local-only

see 2.26.2

2.32 show ((mac address-table) | mac-address-table) mlag-peer diff peer-only

see 2.26.2

2.33 show ((mac address-table) | mac-address-table) multicast [...]

igmpMacEntries	*list of IgmpSnoopingTableEntry* see 2.33.1	All IGMP Snooping MAC entries.
multicastMacTable	*MulticastMacAddressTable* see 2.26.1	Multicast MAC entries.

2.33.1 IgmpSnoopingTableEntry

interfaces	*list of Interface name*	Interfaces of the MAC address.
macAddr	*MAC address*	MAC address.
macType	*igmp*	Type of MAC address.
vlanId	*integer*	VLAN ID of MAC address.

2.34 show ((mac address-table) | mac-address-table) multicast brief [...]

see 2.33

2.35 show ((mac address-table) | mac-address-table) notification host-flap detection window

detectionWindow	*integer*	The detection window for MAC Flapping, in seconds.

2.36 show ((mac address-table) | mac-address-table) notification host-flap logging

status	*boolean*	The current configuration of MAC Flap logging.

2.37 show (arp | (ip arp)) [...]

dynamicEntries	*integer*	Number of dynamically learnt addresses in ARP table.
ipV4Neighbors	*list of IpV4Neighbor* see 2.37.1	IPv4 neighbors discovered by the Address Resolution Protocol.
notLearnedEntries	*integer*	Number of addresses in ARP table not due to learning.
staticEntries	*integer*	Number of statically configured addresses in ARP table.
totalEntries	*integer*	Total number of addresses in ARP table.

2.37.1 IpV4Neighbor

address	*string*	IPv4 address or hostname.
[age]	*integer*	How many minutes ago the neighbor entry was learned.
[extra]	*string*	Extra metadata about this neighbor's discovery.
hwAddress	*string*	Hardware (layer 2) address learnt for the corresponding network (layer 3) address.
interface	*string*	Interface name the neighbor was learned on.

2.38 show (arp | (ip arp)) resolve summary total [...]

see 2.37

2.39 show (arp | (ip arp)) summary [...]

see 2.37

2.40 show (arp | (ip arp)) | (ipv6 neighbors) remote [...]

`vlans`	*table(numeric string) = ArpBindingVlan* see 2.40.1	A mapping of VLAN to its ARP bindings.

2.40.1 ArpBindingVlan

`attributes` **	*table(IPv4 or IPv6 address) = ArpBindingType* see 2.40.1.1	A mapping of IP address to its MAC address attributes. **Since: 4.26.2
`ipAddresses`	*table(IPv4 or IPv6 address) = MAC address*	A mapping of IP address to its MAC address.

2.40.1.1 ArpBindingType

`evpnMultihomed`	*boolean*	The IP address is EVPN multihomed.
`vcsRemote` **	*boolean*	The IP address is received from VCS. **Since: 4.29.1

2.41 show (bgp rpki cache) [...]

caches	*table(string) = CacheModel* see 2.41.1	Collection of cache server information indexed by cache server's name.

2.41.1 CacheModel

activeSince	*floating point number*	Time the connection became active in UTC.
deleted	*boolean*	The config for this cache has been deleted.
[detail]	*CacheDetailModel* see 2.41.1.1	Detail information.
entries	*integer*	Number of Route Origin Authorization entries.
expireInterval	*integer*	Number of seconds to retain data synced from cache server.
host	*string*	IP address or hostname of cache server.
lastFullSync	*floating point number*	Time of last reset sync with cache server in UTC.
lastResetQuery	*floating point number*	Time of last reset query sent to cache server in UTC.
lastSerialQuery	*floating point number*	Time of last serial query sent to cache server in UTC.
lastUpdateSync	*floating point number*	Time of last serial sync with cache server in UTC.
port	*integer*	Transport port of cache server.
preference	*integer*	Cache server preference. Lower value means higher preference.
protocolVersion	*enum: version0 | version1 | notNegotiatedYet*	RPKI-RTR protocol version.
[reasonInactive]	*string*	Reason the connection is not active.
refreshInterval	*integer*	Number of seconds between cache server polls.
retryInterval	*integer*	Number of seconds between poll error and cache server poll.
roaTables **	*list of string*	RPKI ROA tables populated by the cache server. **Since: 4.27.1
serialNumber **	*integer*	Serial number for the last successful sync. **Since: 4.24.2
sessionId **	*integer*	Session ID for this cache instance. **Since: 4.24.2
state	*enum: idle | disconnected | connected | synced | misconfigured | syncing | requestRoaUpdate | disabled | requestRoaReset*	Cache server current state.
[transportInformation] **	*CacheTransportInformationModel* see 2.41.1.2	Transport information. **Since: 4.26.2
transportProtocol **	*enum: tcp | tls*	Transport protocol for connection to RPKI cache server. **Since: 4.25.0

vrf	*string*	VRF where the cache server is configured.

2.41.1.1 CacheDetailModel

lastConfigChange	*floating point number*	Time when the host, port, VRF or local interface was changed in the config in UTC.
lastConnectionError	*floating point number*	Time of last connection error to cache server in UTC.
lastProtocolError	*floating point number*	Time of sending/receiving protocol error to/from cache server in UTC.
overrideConfiguredInterval-Values	*boolean*	Cache suggested or default interval values will be used.
[tcpInformation]	*TcpInformationModel* see 2.41.1.1.1	TCP information.
tcpKeepaliveIdleTime **	*integer*	Idle time (seconds) before TCP keepalive. **Since: 4.25.1
tcpKeepaliveProbeCount **	*integer*	Number of keepalive probes before closing connection. **Since: 4.25.1
tcpKeepaliveProbeInterval **	*integer*	Interval (seconds) between TCP keepalive probes. **Since: 4.25.1

2.41.1.1.1 TcpInformationModel

congestionWindow	*integer*	TCP send congestion window (bytes).
delayedAckTimeout	*integer*	TCP socket delayed ack timeout (microseconds).
inputQueueLength	*integer*	TCP input queue length.
inputQueueMaxLength	*integer*	TCP input queue max length.
localTcpAddress	*IP address*	TCP local address.
localTcpPort	*integer*	TCP local port.
maxSegmentSize	*integer*	TCP outgoing maximum segment size (bytes).
options	*TcpOptionsModel* see 2.41.1.1.1.1	TCP socket options.
outputQueueLength	*integer*	TCP output queue length.
outputQueueMaxLength	*integer*	TCP output queue max length.
recvRtt	*integer*	TCP receive round-trip time (microseconds).
recvWindow	*integer*	TCP advertised receive window (bytes).
recvWindowScale	*integer*	TCP socket receive window scale factor.
retransmitTimeout	*integer*	TCP socket retransmission timeout (microseconds).
sendRtt	*integer*	TCP round-trip time (microseconds).
sendRttVariance	*integer*	TCP round-trip time variance (microseconds).
sendWindowScale	*integer*	TCP socket send window scale factor.
slowStartThreshold	*integer*	TCP send slow start size threshold (bytes).

state	*enum: CLOSING | NOTUSED | LISTEN | SYN-RECEIVED | CLOSE-WAIT | FIN-WAIT-2 | CLOSED | LAST-ACK | ESTABLISHED | TIME-WAIT | FIN-WAIT-1 | SYN-SENT*	TCP Socket current state.
totalRetrans	*integer*	Total number of TCP retransmissions.

2.41.1.1.1.1 TcpOptionsModel

ecn	*boolean*	Explicit Congestion Notification enabled.
selectiveAcks	*boolean*	Selective Acknowledgments enabled.
timestamps	*boolean*	Timestamps enabled.
windowScale	*boolean*	Window Scale enabled.

2.41.1.2 CacheTransportInformationModel

[sslProfileName]	*string*	Configured SSL profile name.

2.42 show (bgp rpki cache counters) [...]

caches	*table(string) = CacheCounterModel* see 2.42.1	A mapping of cache names to their counters.

2.42.1 CacheCounterModel

cacheReset	*integer*	Count of Cache Reset PDU.
cacheResponse	*integer*	Count of Cache Response PDU.
endOfData	*integer*	Count of End of Data PDU.
ipv4Prefix	*integer*	Count of IPv4 Prefix PDU.
ipv6Prefix	*integer*	Count of IPv6 Prefix PDU.
resetQuery	*integer*	Count of Reset Query PDU.
serialNotify	*integer*	Count of Serial Notify PDU.
serialQuery	*integer*	Count of Serial Query PDU.

2.43 show (bgp rpki cache counters) errors [...]

cachesRx	*table(string) = CacheRxErrorCounterModel* see 2.43.1	A mapping of cache names to their received error counters.
cachesTx	*table(string) = CacheTxErrorCounterModel* see 2.43.2	A mapping of cache names to their sent error counters.

2.43.1 CacheRxErrorCounterModel

corruptData	*integer*	Count of Corrupt Data PDU.
internalError	*integer*	Count of Internal Error PDU.
invalidRequest	*integer*	Count of Invalid Request PDU.
noData	*integer*	Count of No Data PDU.
other	*integer*	Count of Other PDU.
pduTimeout	*integer*	Count of timeouts waiting for PDU.
unexpectedProtocol **	*integer*	Count of Unexpected Protocol PDU. **Since: 4.24.1
unsupportedPdu	*integer*	Count of Unsupported PDU.
unsupportedProtocol	*integer*	Count of Unsupported Protocol PDU.

2.43.2 CacheTxErrorCounterModel

corruptData	*integer*	Count of Corrupt Data PDU.
duplicateAnnouncement	*integer*	Count of Duplicate Announcement PDU.
internalError	*integer*	Count of Internal Error PDU.
other	*integer*	Count of Other PDU.
unexpectedProtocol **	*integer*	Count of Unexpected Protocol PDU. **Since: 4.24.1
unsupportedPdu	*integer*	Count of Unsupported PDU.
unsupportedProtocol	*integer*	Count of Unsupported Protocol PDU.
withdrawalUnknown	*integer*	Count of Withdrawal Unknown PDU.

2.44 show (cli | routing-context) vrf

vrfRoutingContext	*string*	Name of VRF for the current routing-context.

2.45 show (ip | ipv6) attached-host route export [...]

revision 2

vrfs	*table(string) = AttachedHostVlan* see 2.45.1	A mapping of VRF to vlan information.

2.45.1 AttachedHostVlan

`vlans`	*table(string) = AttachedHostList* see 2.45.1.1	A mapping of vlan to hosts information.

2.45.1.1 AttachedHostList

`hosts`	*list of AttachedHostEntry* see 2.45.1.1.1	List of all the injected hosts.

2.45.1.1.1 AttachedHostEntry

`address`	*IP address*	IP Address.
`interface`	*interface name*	Interface on which host is attached to.
`prefix` **	*IP address prefix*	Injected prefix. **Since: 4.22.0

2.46 show (ip | ipv6) bgp summary lldp [...]

revision 3

vrfs	*table(string) = BgpSummary* see 2.46.1	Per VRF BGP summary.

2.46.1 BgpSummary

asn	*string*	Autonomous System Number.
peers	*table(string) = BgpSummaryPeerEntry* see 2.46.1.1	Dictionary of BGP peer entries indexed by the peer address.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF Name.

2.46.1.1 BgpSummaryPeerEntry

asn	*string*	Autonomous System Number.
[description] **	*string*	Peer Description. **Since: 4.21.3
inMsgQueue	*integer*	Number of messages waiting to be processed.
[lldpNeighbors] **	*list of string*	LLDP Neighbor list (only multi-agent mode). **Since: 4.27.2
msgReceived	*integer*	Number of messages received.
msgSent	*integer*	Number of messages sent.
outMsgQueue	*integer*	Number of messages waiting to be sent.
peerState	*enum: OpenConfirm | NotNegotiated | Active | Connect | Idle | Established | OpenSent*	Current peer state. Idle - Ignoring everything Connect - Trying to connect Active - Waiting for a connection OpenSent - Open packet has been sent OpenConfirm - Waiting for a keepalive or notify Established - Connection has been established.

[peerStateIdleReason]	*enum: MaxAdvertisedPath | OrrPositionIncomplete | OrrLinkLocalPeer | OrrPositionCountExceeds-Maximum | NoRemoteAs | AsuShutdown | UnsupportedCaps | MinHoldTime | NoLocalNexthops | PendingNotification | GroupAdmin | IbgpLinkLocalUnsupported | OrrPositionPeerAddress-DynamicPeer | NoInterface | MaxPath | Deleted | Damping | Other | NoActivate | MaxAcceptedPath | Admin | OrrAndAddPathBackup | EbgpLinkLocalMultihop-Unsupported*	Used when peer state is "Idle". NoInterface - The peer's interface is down Admin - The peer is disabled GroupAdmin - The peer's group is disabled MaxPath - The peer has exceeded its max route limit Deleted - The peer is deleted and waiting for deffered cleanup NoActivate - The peer is not activated in any address-family mode NoLocalNexthops - The peer is missing local addresses for activated AFI/SAFIs UnsupportedCaps - The peer has not negotiated required capabilities NoRemoteAs - Missing remote-as for peer AsuShutdown - ASU2 shutdown in progress IbgpLinkLocalUnsupported - IBGP peering over IPv6 link-local address not supported MaxAcceptedPath - The peer has exceeded its max accepted route limit Other - The peer is idled for some other reason PeerNotification - The peer is pending notification message to be sent EbgpLinkLocalMultihopUnsupported - EBGP multihop session over IPv6 link-local addresses not supported MaxAdvertisedPath - The peer has exceeded its max advertised route limit Damping - The peer is waiting for BFD connection to be stable.
prefixAccepted	*integer*	Number of prefixes accepted.
[prefixInBest] **	*integer*	Number of paths select as best. **Since: 4.22.0
[prefixInBestEcmp] **	*integer*	Number of paths end in best ecmp group. **Since: 4.22.0
prefixReceived	*integer*	Number of prefixes receieved from peer.
[rulesAccepted] **	*integer*	Number of flowspec rules accepted from peer. **Since: 4.21.3
[rulesReceived] **	*integer*	Number of flowspec rules received from peer. **Since: 4.21.3
[underMaintenance]	*boolean*	Peering under maintenance.
upDownTime	*floating point number*	Timestamp of last up/down transition.
version	*integer*	BGP version.

2.47 show (ip | ipv6) hardware ale routes acknowledgement [...]

[cmdErr]	*string*	Error Message.
fibRoutes	*table(IPv4 or IPv6 address prefix) = Acknowledgement-RouteEntry* see 2.47.1	Mapping prefix to its FibRoute Info.
[vrfName]	*string*	VRF name.

2.47.1 AcknowledgementRouteEntry

[acknowledged]	*AcknowledgementRoute-VersionId* see 2.47.1.1	Acknowledged Route Version Id.
[requested]	*AcknowledgementRoute-VersionId* see 2.47.1.1	Requested Route Version Id.

2.47.1.1 AcknowledgementRouteVersionId

failed **	*boolean*	Route install failed. **Since: 4.28.0
requestedDelete	*boolean*	Route due to be deleted.
versionId	*integer*	Route Version Id to be acknowledged.

2.48 show (ip | ipv6) hardware ale routes drop [...]

routingTelemetry **	*enum: disabled | enabled*	Routing telemetry status. **Since: 4.29.2
[vrfDropRoutes]	*list of VrfDropRoute* see 2.48.1	List of all drop routes.

2.48.1 VrfDropRoute

insertTime	*floating point number*	Drop route creation time.
prefix	*IP address prefix*	The route's prefix.
vrfName	*string*	Name of a VRF.

2.49 show (ip | ipv6) hardware ale routes inconsistent [...]

routingTelemetry **	*enum: disabled | enabled*	Routing telemetry status. **Since: 4.29.2
[vrfInconsistentRoutes]	*list of VrfRouteEntry* see 2.49.1	List of all inconsistent routes.

2.49.1 VrfRouteEntry

insertTime	*floating point number*	Route entry creation time.
prefix	*IP address prefix*	The route's prefix.
vrfName	*string*	Name of a VRF.

2.50 show (ip | ipv6) hardware fib fec [...]

[cmdErr]	*string*	Error flag.
[compressedByRoute] **	*IP address prefix*	IP Address of the parent route. **Since: 4.22.0
[fec]	*table(numeric string) = FibFecInfo* see 2.50.1	Mapping fecId to its FEC Info.
[routeCompressed] **	*boolean*	Route is compressed. **Since: 4.22.0
[routeType]	*enum: multicast | connected | ebgp | directedBcast | bgp | isisL2 | rip | receiveBcast | routeCacheForward | ibgpLu | l2NexthopGroup | staticEosSdk | vcs | staticConfig | redirect | dhcp | lbgp | ospfNssaE2 | ibgp | ospfv3NssaE2 | attached | ospfv3NssaE1 | ospfv3 | ospfNssaE1 | ospfAggregate | ospfv3AseE1 | isisL1 | nexthopGroup | ospfv3AseE2 | ospfIntraArea | receiveAndHostRoute | ospfAseE1 | routeCacheConnected | kernel | ospf | martian | unknown | ospfInterArea | dynamicPolicy | bgpAggregate | gribi | cbfLeaked | drop | policy | invalid | ospfAseE2 | ebgpLu | receive*	Route type.

2.50.1 FibFecInfo

[backupFecVias] **	*list of AleL3AdjVia* see 2.50.1.1	Backup Via list. **Since: 4.23.2
[cmdErr]	*string*	Error flag.

[fecType]	*enum: usedByTunnelRoute | attachedRoute | directedBcastRoute | tunnelFibRoute | dropRoute | receiveRoute | routeCacheForwardRoute | evpnRoute | routeCacheConnectedRoute | routeTypeUnknown | invalidRoute | policyRoute | fecInfoUnavailable | nexthopRoute | mplsRoute | mplsFloodSetRoute | bcastReceiveRoute | l2NexthopGroupRoute | nexthopGroupRoute | forwardRoute | vplsMcastRoute | slowReceiveRoute | pwRoute | vplsL2UcastRoute | mplsMcastRoute | greMcastRoute*	FEC type.
[fecVias]	*list of AleL3AdjVia* see 2.50.1.1	Via list.
[vias]	*integer*	Number of vias for the route.

2.50.1.1 AleL3AdjVia

[fallbackInfo] **	*FallbackInfo* see 2.50.1.1.2	Fallback information. **Since: 4.27.2
l3IntfId	*interface name*	L3 Interface Id.
nextHopIp	*IP address prefix*	Ip Address of Next Hop.
[tunnelDescriptor] **	*AleL3AdjViaTunnel* see 2.50.1.1.1	Tunnel information. **Since: 4.21.0
weight	*integer*	Metric representing the hop cost.

2.50.1.1.1 AleL3AdjViaTunnel

[tunnelAddressFamily]	*enum: IPv6 | IPv4*	Address family.
tunnelIndex	*integer*	Tunnel table index.
tunnelType	*string*	Tunnel type.

2.50.1.1.2 FallbackInfo

actionType	*enum: none | ipip*	Decapsulation type.
fallbackVrf	*string*	Fallback VRF name.

2.51 show (ip | ipv6) hardware fib summary

revision 2

[adjResourceOptimization] **	*AdjResourceOptimization* see 2.51.1	Adjacency resource optimization. **Since: 4.23.2
[adjShare]	*enum: disabled | enabled*	Adjacency sharing status.
[bfdEvent]	*enum: disabled | enabled*	BFD peer event.
deletionDelay	*floating point number*	Deletion delay.
fibCompression **	*enum: disabled | enabled*	FIB compression status. **Since: 4.22.0
icmpStatus	*enum: disabled | enabled*	ICMP status.
[maxAleEcmp] **	*integer*	Max Ale ECMP. **Since: 4.21.3
maxFecHierarchyLevels **	*integer*	Maximum FEC hierarchy levels. **Since: 4.29.1
maxRoutes	*integer*	Maximum number of routes.
[pbrSupport]	*enum: supported | notSupported*	PBR support.
protectDefaultRoute	*enum: disabled | enabled*	Protect default route status.
[ucmpWeightDeviation] **	*floating point number*	UCMP weight deviation. **Since: 4.21.3
urpfSupport	*enum: supported | notSupported*	URPF support.

2.51.1 AdjResourceOptimization

[enabled]	*enum: disabled | enabled*	Adjacency resource optimization enabled.
supported	*enum: supported | notSupported*	Adjacency resource optimization support.
[thresholds]	*AdjResourceOptimization-Thresholds* see 2.51.1.1	Adjacency resource optimization thresholds.

2.51.1.1 AdjResourceOptimizationThresholds

high	*integer*	Threshold for stopping resource optimization.
low	*integer*	Threshold for starting resource optimization.

2.52 show (ip | mac) access-lists payload alias

udfAliases	*table(string) = PayloadSpec* see 2.52.1	Mapping of alias name to payload.

2.52.1 PayloadSpec

headerStart **	*boolean*	True, if offset is from header start. **Since: 4.15.3
payload	*list of PayloadElement* see 2.52.1.1	List of offset,pattern,mask tuple.

2.52.1.1 PayloadElement

`alias` **	*string*	Alias of payload. **Since: 4.20.0
`maskOverride` **	*boolean*	True, if mask is overridden in alias. **Since: 4.21.0
`offset`	*integer*	Offset inside payload for DPI.
`pattern`	*integer*	Pattern to match at offset inside payload.
`patternMask`	*integer*	Mask for the pattern to match at offset inside payload.
`patternOverride` **	*boolean*	True, if pattern is overridden in alias. **Since: 4.21.0

2.53 show ADDR_FAMILY dhcp relay access-list [...]

[ipAclList]	*IpAclList* see 2.53.1	IP access list.
[ipv6AclList]	*Ipv6AclList* see 2.53.2	IPv6 access list.

2.53.1 IpAclList

aclList	*list of _AclBase* see 2.53.1.1	List of ACLs.

2.53.1.1 _AclBase

Could also be an instance of:
AclSummary 2.53.1.2
ServiceAclSummary 2.53.1.3

countersEnabled	*boolean*	True if counters are enabled.
countersIncomplete	*boolean*	True if the hardware could not allocate counters for every rule.
dynamic **	*boolean*	True if a dynamic (non-persistent) access control list. **Since: 4.21.3
name	*string*	Name.
readonly	*boolean*	True if read-only.
standard	*boolean*	True if a standard access control list - standard ACLs filter only on the source address.
type	*enum: MacAcl | Ip6Acl | Ip4Acl*	Type of ACL.

2.53.1.2 AclSummary

activeEgressIntfs	*list of IntfSummary* see 2.53.1.2.1	Active egress interfaces.
activeIngressIntfs	*list of IntfSummary* see 2.53.1.2.1	Active ingress interfaces.
configuredEgressIntfs	*list of IntfSummary* see 2.53.1.2.1	Configured egress interfaces.
configuredIngressIntfs	*list of IntfSummary* see 2.53.1.2.1	Configured ingress interfaces.
countersEnabled	*boolean*	True if counters are enabled.
countersIncomplete	*boolean*	True if the hardware could not allocate counters for every rule.
dynamic **	*boolean*	True if a dynamic (non-persistent) access control list. **Since: 4.21.3
name	*string*	Name.
notFullySupportedByHardware	*boolean*	True if some rules are not fully supported by hardware.
readonly	*boolean*	True if read-only.

sequenceLength	*integer*	Total number of rules and remarks in the list.
standard	*boolean*	True if a standard access control list - standard ACLs filter only on the source address.
type	*enum: MacAcl | Ip6Acl | Ip4Acl*	Type of ACL.

2.53.1.2.1 IntfSummary

name	*string*	Name of interface.
[target] **	*string*	Where the ACL is applied. **Since: 4.18.0
[vrf]	*string*	VRF name, since the summary is per vrf.

2.53.1.3 ServiceAclSummary

activeVrfs	*list of string*	Active on vrfs.
configuredVrfs	*list of string*	Configured on vrfs.
countersEnabled	*boolean*	True if counters are enabled.
countersIncomplete	*boolean*	True if the hardware could not allocate counters for every rule.
dynamic **	*boolean*	True if a dynamic (non-persistent) access control list. **Since: 4.21.3
name	*string*	Name.
readonly	*boolean*	True if read-only.
sequenceLength	*integer*	Total number of rules and remarks in the list.
standard	*boolean*	True if a standard access control list - standard ACLs filter only on the source address.
type	*enum: MacAcl | Ip6Acl | Ip4Acl*	Type of ACL.

2.53.2 Ipv6AclList

aclList	*list of _AclBase* see 2.53.1.1	List of ACLs.

2.54 show ADDR_FAMILY dhcp relay counters

globalCounters	*GlobalCounters* see 2.54.1	DHCP relay global counters.
[interfaceCounters]	*table(Interface name) = InterfaceCounter* see 2.54.2	A mapping of interface to its counters.

2.54.1 GlobalCounters

`allRequests`	*Counters* see 2.54.1.1	Global counter for requests.
`allResponses`	*Counters* see 2.54.1.1	Global counter for responses.
`lastResetTime`	*floating point number*	Last reset time for global counters.

2.54.1.1 Counters

`dropped`	*integer*	Dropped packets.
`forwarded`	*integer*	Forwarded packets.
`received`	*integer*	Received packets.

2.54.2 InterfaceCounter

`lastResetTime`	*floating point number*	Last reset time for interface counter.
`replies`	*Counters* see 2.54.1.1	Interface counter for replies.
`requests`	*Counters* see 2.54.1.1	Interface counter requests.

2.55 show ADDR_FAMILY helper-address

revision 2

activeState	*boolean*	DHCP relay is active.
allSubnets	*boolean*	DHCP all-subnets relaying is enabled.
allSubnetsV6	*boolean*	DHCPv6 all-subnets relaying is enabled.
alwaysOn	*boolean*	DHCP relay alwaysOn state.
configuredInterfaces	*table(Interface name) = ConfiguredInterface* see 2.55.1	A mapping of interface to its DHCP configuration.
[defaultServers]	*DefaultServersShowModel* see 2.55.2	DHCP Relay mode default addresses/hostnames.
dhcpReqFloodRestrictDisabledDueToSnooping **	*boolean*	DHCP client request flooding suppression disbled due to DHCP snooping configuration. **Since: 4.30.1
linkLayerAddrOpt	*boolean*	DHCPv6 relay link-layer address option (79) is enabled.
mlagPeerLinkReqDisable **	*boolean*	DHCP requests received over MLAG peer link are suppressed. **Since: 4.27.2
option82	*boolean*	DHCP relay information option (82) is enabled.
remoteIdEncodingFormat	*string*	DHCPv6 relay remote ID option (37) encoding format.
reqFloodSuppressionVlanConfig **	*list of integer*	DHCP client request flooding suppressed VLANs configured. **Since: 4.30.1
reqFloodSuppressionVlanOperational **	*list of integer*	DHCP client request flooding suppressed VLANs operational. **Since: 4.30.1
tunnelReqDisable	*boolean*	DHCP tunnel requests are disabled.
[vssSubOptUnsupportedServers] **	*VssUnsupportedServersModel* see 2.55.3	DHCP VSS control sub-option (152) and DHCPv6 VSS option (68) unsupported servers. **Since: 4.29.0

2.55.1 ConfiguredInterface

allSubnets	*boolean*	DHCP all-subnets relaying enabled interface.
allSubnetsV6	*boolean*	DHCPv6 all-subnets relaying enabled interface.
circuitId	*string*	Circuit ID enabled interface.
disabledV4	*boolean*	DHCPv4 relay is disabled.
disabledV6	*boolean*	DHCPv6 relay is disabled.
[helpers]	*list of Helper* see 2.55.1.1	Helper addresses/hostnames for an interface.
usingGlobalModeServersV4	*boolean*	Using DHCPv4 Relay mode configurations.
usingGlobalModeServersV6	*boolean*	Using DHCPv6 Relay mode configurations.

2.55.1.1 Helper

[linkAddr]	*IPv6 address*	Link address.
[serverAddr]	*IP address*	Helper address.
[serverHostname]	*string*	Helper hostname.
[srcAddr]	*IP address*	Source address.
[srcIntf]	*interface name*	Source interface.
vrf	*string*	Server VRF.

2.55.2 DefaultServersShowModel

helpersV4	*list of Helper* see 2.55.1.1	DHCP Relay mode default IPv4 addresses/hostnames.
helpersV6	*list of Helper* see 2.55.1.1	DHCP Relay mode default IPv6 addresses.

2.55.3 VssUnsupportedServersModel

vssHelpersV4	*list of Helper* see 2.55.1.1	IPv4 DHCP Servers not supporting VSS sub-option.
vssHelpersV6	*list of Helper* see 2.55.1.1	IPv6 DHCP Servers not supporting VSS sub-option.

2.56 show APPLICATION profile

instances	*table(string) = FpgaApplicationInstanceInfo* see 2.56.1	A mapping of FPGA application instance to information specific to each application instance.

2.56.1 FpgaApplicationInstanceInfo

attributes **	*table(string) = string*	Profile attributes. **Since: 4.30.1
fpgaId	*string*	FPGA ID running the application.
[profile]	*string*	Profile Name.
[reason]	*string*	Reason for the error (if in error state).
state	*enum: applying | error | unknown | applied*	State of this profile instance.

2.57 show APPLICATION profile all

profiles	*table(string) = FpgaApplicationProfile* see 2.57.1	Profiles available for the application on the FPGA.

2.57.1 FpgaApplicationProfile

`attributes` **	*table(string) = string*	Profile attributes. **Since: 4.30.1
`compatibleFpgas`	*list of string*	FPGAs that this profile is compatible with.
`description`	*string*	Profile description.
`name`	*string*	Profile name.

2.58 show GNSS status

gnss	*table(string) = Gnss* see 2.58.1	Map GNSS names to their specifications.

2.58.1 Gnss

altitude	*floating point number*	Altitude.
altitudeOffset	*floating point number*	Offset of Altitude.
clockNsec	*integer*	Nanoseconds from the system clock.
clockSec	*integer*	Seconds from the system clock.
latitude	*floating point number*	Latitude.
latitudeOffset	*floating point number*	Offset of Latitude.
longitude	*floating point number*	Longitude.
longitudeOffset	*floating point number*	Offset of Longitude.
mode	*enum: threeD | noFix | twoD | unknownMode*	Specification for GNSS positioning and grading.
ppsAvailable	*boolean*	PPS available.
ppsNsec	*integer*	Nanoseconds from the PPS source.
ppsSec	*integer*	Seconds from the PPS source.
quantizationErr	*integer*	Quantization error.
satellites	*table(string) = Satellite* see 2.58.1.1	Map satellite names to their specifications.
timeOffset	*floating point number*	Offset of unix time seconds.
unixTimeNsec	*integer*	Nano Seconds past the most recent Seconds since epoch.
unixTimeSec	*integer*	Seconds since epoch.
version	*Version* see 2.58.1.2	Versions of GNSS.

2.58.1.1 Satellite

azimuth	*floating point number*	Rotation (degrees) of the whole antenna around a vertical axis.
elevation	*floating point number*	Angle (degrees) between satellite and horizontal plane.
frequencyId	*integer*	Frequency Id of Satellite 0-13.
gnssId	*integer*	Type of GNSS.
health	*enum: healthy | unknownHealth | unhealthy*	Specification for the health of the satellite.
inUse	*boolean*	Whether the current satellite is in use.
name	*string*	Name of Satellite.
pseudorandomNoiseId	*integer*	Signal that looks as if it is a random noise signal, but actually repeats after a certain length.
snr	*floating point number*	Signal to noise ratio.
spaceVehicleId	*integer*	Serial number assigned to each GPS satellite.

2.58.1.2 Version

`firmwareRevision`	*string*	Name of the firmware revision of GNSS.
`firmwareVersion`	*string*	Name of the firmware version of GNSS.
`modelName`	*string*	Model name of the GNSS.
`serialNumber`	*string*	Serial Number of GNSS.

2.59 show SYNCE

`[info]`	*SynceStatusInfo* see 2.59.1	Synchronous Ethernet status.

2.59.1 SynceStatusInfo

`[cascadedEecCount]` **	*integer*	Number of cascaded EECs. **Since: 4.29.1
`[cascadedEeecCount]` **	*integer*	Number of cascaded eEECs. **Since: 4.29.1
`[clockIdentity]` **	*string*	Synce clockIdentity of the originator of the extended QlTlv. **Since: 4.29.1
`[etlvFlag]` **	*integer*	EEC/eEEC flag. **Since: 4.29.1
`[freeRunningAndHoldover-QualityLevel]` **	*enum: QL-PRC* | *QL-FAILED* | *QL-EPRC* | *QL-PRS* | *QL-ST2* | *QL-SMC* | *QL-SSU-B* | *QL-INVX* | *QL-EPRTC* | *QL-STU* | *QL-ST3E* | *QL-EEC1* | *QL-UNC* | *QL-NSUPP* | *QL-EEC2* | *QL-DNU* | *QL-PROV* | *QL-ESEC* | *QL-INV* | *QL-TNC* | *QL-DUS* | *QL-PRTC* | *QL-SSU-A*	Quality level of the device when operating in either free-running or holdover state. **Since: 4.28.1
`holdoffDuration` **	*floating point number*	Duration of holdoff timer in seconds. **Since: 4.28.1
`[networkOption]`	*enum: 1* | *2*	Network option.
`operState`	*enum: lockedAndHoldover-Acquired* | *locked* | *holdover* | *freeRunning*	Operational state of the device.
`qualityLevelEnabled` **	*boolean*	Quality level processing enabled. **Since: 4.28.1
`[selectedClock]`	*ClockSource* see 2.59.1.1	Selected clock source to which the deviced is synchronized.

[selectedQualityLevel]	*enum: QL-PRC | QL-FAILED | QL-EPRC | QL-PRS | QL-ST2 | QL-SMC | QL-SSU-B | QL-INVX | QL-EPRTC | QL-STU | QL-ST3E | QL-EEC1 | QL-UNC | QL-NSUPP | QL-EEC2 | QL-DNU | QL-PROV | QL-ESEC | QL-INV | QL-TNC | QL-DUS | QL-PRTC | QL-SSU-A*	Quality level of the selected clock source.
waitToRestoreDuration **	*floating point number*	Duration of wait-to-restore timer in seconds. **Since: 4.28.1

2.59.1.1 ClockSource

[ethernet] **	*interface name*	Ethernet interface providing this clock. **Since: 4.28.1

2.60 show aaa

[enablePassword]	*string*	The encrypted enable password.
users	*table(string) = AaaUser* see 2.60.1	Maps a username to details for this user.

2.60.1 AaaUser

password	*string*	The encrypted password of this account.

2.61 show aaa (methods | method-lists) accounting

commandsAcctMethods	*table(string) = AaaMethodListsAcctMethods* see 2.61.1	A Mapping between command accounting type and methods configured.
dot1xAcctMethods	*table(string) = AaaMethodListsAcctMethods* see 2.61.1	A Mapping between dot1x accounting type and methods configured.
execAcctMethods	*table(string) = AaaMethodListsAcctMethods* see 2.61.1	A Mapping between exec accounting type and methods configured.
systemAcctMethods	*table(string) = AaaMethodListsAcctMethods* see 2.61.1	A Mapping between system accounting type and methods configured.

2.61.1 AaaMethodListsAcctMethods

[consoleAction]	*string*	Present if accounting for console actions is enabled.
consoleMethods	*list of string*	List of methods for a console accounting.

[defaultAction]	*string*	Present if accounting for default actions is enabled.
defaultMethods	*list of string*	List of methods for default accounting.

2.62 show aaa (methods | method-lists) all

accounting	*AaaMethodListsAcct* see 2.61	AAA Accounting information.
authentication	*AaaMethodListsAuthen* see 2.62.1	AAA Authentication information.
authorization	*AaaMethodListsAuthz* see 2.62.2	AAA Authorization information.

2.62.1 AaaMethodListsAuthen

dot1xAuthenMethods	*table(string) = AaaMethodListsMethods* see 2.62.1.1	A Mapping between dot1x authentication type and methods configured.
enableAuthenMethods	*table(string) = AaaMethodListsMethods* see 2.62.1.1	A Mapping between enable authentication type and methods configured.
loginAuthenMethods	*table(string) = AaaMethodListsMethods* see 2.62.1.1	A Mapping between login authentication type and methods configured.

2.62.1.1 AaaMethodListsMethods

methods	*list of string*	List of methods for a particular mode.

2.62.2 AaaMethodListsAuthz

commandsAuthzMethods	*table(string) = AaaMethodListsMethods* see 2.62.1.1	A Mapping between command authorization type and methods configured.
execAuthzMethods	*table(string) = AaaMethodListsMethods* see 2.62.1.1	A Mapping between exec authorization type and methods configured.

2.63 show aaa (methods | method-lists) authentication

see 2.62.1

2.64 show aaa (methods | method-lists) authorization

see 2.62.2

2.65 show aaa authentication lockout

lockedUsers	*table(string) = AaaLockedUser* see 2.65.1	Maps a username to details for this user.

2.65.1 AaaLockedUser

lockoutEndTime	*floating point number*	Time that user's account will be unlocked.
lockoutStartTime	*floating point number*	Time that user's account was locked.

2.66 show aaa counters

accountingError	*integer*	Number of error accounting requests.
accountingSuccess	*integer*	Number of successful accounting requests.
authenticationFail	*integer*	Number of failed authentication requests.
authenticationSuccess	*integer*	Number of successful authentication requests.
authenticationUnavailable	*integer*	Number of unavailable authentication requests.
authorizationAllowed	*integer*	Number of allowed authorization requests.
authorizationDenied	*integer*	Number of denied authorization requests.
authorizationUnavailable	*integer*	Number of unavailable authorization requests.
[counterResetTimestamp]	*floating point number*	Timestamp of last counter reset.
pendingAccountingRequests	*integer*	Number of pending accounting requests.

2.67 show aaa sessions

nonInteractives	*table(numeric string) = Sessions* see 2.67.1	Session information for nonInteractive connections.
serials	*table(numeric string) = Sessions* see 2.67.1	Session information for serial connections.
vtys	*table(numeric string) = Sessions* see 2.67.1	Session information for ssh or telnet connections.

2.67.1 Sessions

authMethod	*string*	Authentication method of the session.
[remoteAddress]	*string*	Remote ip address.
[remoteHost]	*string*	Remote hostname.
role	*string*	Role assigned to the user.
[service]	*string*	Service.
sessionStartTime	*integer*	UTC time at which session was started.
state	*enum: pending | established*	State of the session either established or pending.

[terminal]	*string*	Maps to terminal type and terminal number.
username	*string*	Login name of the user of this session.

2.68 show adaptive-virtual-topology application-profile [...]

vrfs	*table(string) = AvtAppProfileByVrfModel* see 2.68.1	Application profile information for a VRF.

2.68.1 AvtAppProfileByVrfModel

appProfiles	*table(string) = AvtProfileByAvtModel* see 2.68.1.1	Application Profile information for an adaptive topolgy.
vni	*integer*	VxLAN network identifier for VRF.
vrf	*string*	VRF name.

2.68.1.1 AvtProfileByAvtModel

appProfileId	*integer*	Application profile ID.
appProfileName	*string*	Application profile name.
avtId	*integer*	Adaptive virtual topology service id.
avtName	*string*	Adaptive virtual topology service name.
state	*string*	If the setting is active in status.

2.69 show adaptive-virtual-topology flow [...]

flows	*list of AvtFlowInfo* see 2.69.1	List of flows.

2.69.1 AvtFlowInfo

avtId	*integer*	Adaptive virtual topology identifier.
[avtName]	*string*	Name of adaptive virtual topology.
[directPathId]	*integer*	Direct path ID.
[egressIntf]	*interface name*	Flow egress interface.
[egressPathName]	*string*	Name of the egress path.
key	*AvtFlowKey* see 2.69.1.1	Flow key.
[labelStack]	*list of AvtSegmentLabel* see 2.69.1.2	Label stack of the multihop path.
[multihopPathId]	*integer*	Multihop path ID.
[nexthopVtepAddr]	*IP address*	Nexthop VTEP IP address.
transit	*boolean*	Transit node.

2.69.1.1 AvtFlowKey

dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.
ipProtocolNumber	*integer*	IP protocol number.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.
vrf	*string*	VRF name.

2.69.1.2 AvtSegmentLabel

label	*integer*	Segment label.

2.70 show adaptive-virtual-topology path [...]

[regionId] **	*integer*	Adaptive virtual topology region ID. **Since: 4.30.1
[regionName] **	*string*	Adaptive virtual topology region name. **Since: 4.30.1
role **	*enum: zone transit | edge | unknown | pathfinder | region transit*	Role of the node. **Since: 4.30.1
[routerId] **	*IP address*	Router address. **Since: 4.30.1
[siteId] **	*integer*	Adaptive virtual topology site ID. **Since: 4.30.1
[siteName] **	*string*	Adaptive virtual topology site name. **Since: 4.30.1
vrfs	*table(string) = AvtPathsByVrfModel* see 2.70.1	Adaptive virtual topology path information for a VRF.
[zoneId] **	*integer*	Adaptive virtual topology zone ID. **Since: 4.30.1
[zoneName] **	*string*	Adaptive vurtual topology zone name. **Since: 4.30.1

2.70.1 AvtPathsByVrfModel

avts	*table(string) = AvtPathsByAvtModel* see 2.70.1.1	Path information for an adaptive virtual topology.
vni	*integer*	VxLAN network identifier for VRF.
vrf	*string*	VRF name.

2.70.1.1 AvtPathsByAvtModel

avtApplicationProfiles	*list of string*	Adaptive virtual topology application profiles.
avtId	*integer*	Adaptive virtual topology identifier.
avtPaths	*table(string) = AvtPath* see 2.70.1.1.1	Adaptive virtual topology paths, keyed by path type and ID.

2.70.1.1.1 AvtPath

destination	*IP address*	Destination VTEP address.
[directPathId]	*integer*	Direct path ID.
[egressPathName]	*string*	Name of the egress multihop path.
flags	*AvtPathsFlags* see 2.70.1.1.1.1	Path flags.
[jitter]	*floating point number*	Jitter of path in msecs.
[labelStack]	*list of AvtSegmentLabel* see 2.69.1.2	Label stack of the multihop path.
[latency]	*floating point number*	Latency of path in msecs.
[loss]	*floating point number*	Loss in path in percentage.

[multihopPathId]	*integer*	Multihop path ID.
[nexthopAddr]	*IP address*	Immediate nexthop VTEP address.
[pathMtu]	*integer*	Maximum transmission unit of path in bytes.
[priority]	*integer*	Priority of path.

2.70.1.1.1.1 AvtPathsFlags

active	*boolean*	Active path used by forwarding engine.
directPath	*boolean*	Direct path.
valid	*boolean*	Valid path processed by forwarding engine.

2.71 show adaptive-virtual-topology path counters [...]

vrfs	*table(string) = AvtCountersByVrfModel* see 2.71.1	Summary of packet distribution on a VRF.

2.71.1 AvtCountersByVrfModel

avts	*table(string) = AvtCountersByAvtModel* see 2.71.1.1	Summary of packet distribution on an AVT.

2.71.1.1 AvtCountersByAvtModel

vteps	*table(IPv4 or IPv6 address) = AvtCountersByVtepModel* see 2.71.1.1.1	Summary of packet distribution on a VTEP IP.

2.71.1.1.1 AvtCountersByVtepModel

paths	*table(string) = AvtCounterEntryModel* see 2.71.1.1.1.1	Summary of packet distribution on a path.

2.71.1.1.1.1 AvtCounterEntryModel

flows	*integer*	Number of flows following the same path.
intf	*interface name*	Local interface name for outgoing traffic.
outBytes	*integer*	Number of out bytes following the same path.
outPkts	*integer*	Number of out pkts following the same path.
rate	*floating point number*	Throughput of the outgoing traffic.

2.72 show address locking

`active`	*boolean*	IP Locking active state.
`configuredIpv4Intfs` **	*list of Interface name*	List of interfaces configured with address locking IPv4. **Since: 4.23.1
`configuredIpv6Intfs` **	*list of Interface name*	List of interfaces configured with address locking IPv6. **Since: 4.23.1
`enabledIntfs`	*table(Interface name) = IpLockingIntfEnabled* see 2.72.1	A mapping of interfaces to their IPv4 and IPv6 address locking enabled status.
`[inactiveReason]`	*string*	Reason that IP Locking is inactive.
`ipv4Enforced` **	*boolean*	Locked-address enforcement is enabled for IPv4. **Since: 4.26.2
`ipv6Enforced` **	*boolean*	Locked-address enforcement is enabled for IPv6. **Since: 4.25.1
`retryInterval` **	*integer*	Configured Lease Query Retry Interval. **Since: 4.30.1
`timeout` **	*integer*	Configured Lease Query Timeout. **Since: 4.30.1

2.72.1 IpLockingIntfEnabled

`ipv4Enabled`	*boolean*	Address Locking is enabled on interface for IPv4.
`[ipv4InactiveReason]` **	*string*	Reason that Address Locking is not enabled for IPv4. **Since: 4.23.1
`[ipv4Mode]` **	*string*	IPv4 address locking mode. **Since: 4.29.2
`ipv6Enabled`	*boolean*	Address Locking is enabled on interface for IPv6.
`[ipv6InactiveReason]` **	*string*	Reason that Address Locking is not enabled for IPv6. **Since: 4.23.1
`[ipv6Mode]` **	*string*	IPv6 address locking mode. **Since: 4.29.2

2.73 show address locking counters

`intfCounters`	*table(Interface name) = IntfCounters* see 2.73.1	A mapping of interfaces to their counters.
`serverCounters`	*table(IPv4 or IPv6 address) = ServerCounters* see 2.73.2	A mapping of server addresses to their counters.

2.73.1 IntfCounters

leaseActive	*integer*	Number of lease active packets.
leaseQuery	*integer*	Number of lease query packets.
leaseUnassigned	*integer*	Number of lease unassigned packets.
leaseUnknown	*integer*	Number of lease unknown packets.

2.73.2 ServerCounters

leaseActiveDropped	*integer*	Number of lease active packets that were dropped.
leaseActiveReceived	*integer*	Number of lease active packets that were received.
leaseQuery	*integer*	Number of lease query packets.
leaseUnassignedDropped	*integer*	Number of lease unassigned packets that were dropped.
leaseUnassignedReceived	*integer*	Number of lease unassigned packets that were received.
leaseUnknownDropped	*integer*	Number of lease unknown packets that were dropped.
leaseUnknownReceived	*integer*	Number of lease unknown packets that were received.
unknown	*integer*	Number of packets dropped due to unrecognized packet type.

2.74 show address locking servers

dhcpV4Servers	*table(IPv4 or IPv6 address) = IpLockingProtocol* see 2.74.1	Mapping of servers queried for DHCP v4 leases to the protocol used to query them.
dhcpV6Servers **	*table(IPv4 or IPv6 address) = IpLockingProtocol* see 2.74.1	Mapping of servers queried for DHCP v6 leases to the protocol used to query them. **Since: 4.25.2

2.74.1 IpLockingProtocol

protocol	*enum: arIpQuery | leaseQuery | arIpv6Query*	Protocol used to query server.

2.75 show address locking table ipv4 [...]

intfLeases	*table(Interface name) = IntfLeases* see 2.75.1	A mapping of an interface to its leases.

2.75.1 IntfLeases

leases	*table(IPv4 or IPv6 address) = Lease* see 2.75.1.1	A mapping of a lease's IP address to its information.

2.75.1.1 Lease

clientMac	*MAC address*	Client's MAC address.
expirationTime	*integer*	Lease expiration time in UTC.
installed	*boolean*	This lease is installed in hardware.
source **	*enum: config | server*	lease source type. **Since: 4.23.2

2.76 show agent AGENT activity

activityLoopExecNsTime	*integer*	UTC time in nanoseconds at which the current activity loop started.
activityMgrLoopCounter	*integer*	Number of activity loops executed.
attrlogNotEnoughWritable-Bytes	*integer*	Number of times FlowOut could not be processed due to no space left in write buffer or FlowOut Quantum is reached.
avgClockNotifTime	*floating point number*	Average clock notification time in seconds.
avgCombinedFlowOutRunTime	*floating point number*	Average run time spent for all FlowOuts in one activity loop in seconds.
avgFdsPerCycle	*floating point number*	Average file descriptors processed per activity loop.
avgFlowOutRunTime	*floating point number*	Average run time spent for an individual FlowOut in one activity loop in seconds.
avgIOWaitTime	*floating point number*	Average IO wait time in seconds.
avgMountTime	*floating point number*	Average mount time in seconds.
avgRemoteMountTime	*floating point number*	Average remote mount time in seconds.
avgTimeBetweenActivityLoops	*floating point number*	Average time between activity loops in seconds.
currentFlowOutCount	*integer*	Current number of FlowOut.
maxClockNotifTime	*floating point number*	Maximum clock notification time in seconds.
maxCombinedFlowOutRunTime	*floating point number*	Maximum run time spent for all FlowOuts in one activity loop in seconds.
maxFdsPerCycle	*integer*	Maximum file descriptors processed per activity loop.
maxFlowOutCount	*integer*	Maximum number of FlowOut in one activity loop.
maxFlowOutRunTime	*floating point number*	Maximum run time spent for an individual FlowOut in one activity loop in seconds.
maxIOWaitTime	*floating point number*	Maximum IO wait time in seconds.
maxMountStatMountName	*string*	Mount name of mount that took longest to process.
maxMountStatPeerName	*string*	Peer name of mount that took longest to process.
maxMountStatTime	*floating point number*	Maximum time spent processing a mount in seconds.
maxRemoteMountStatMountName	*string*	Mount name of remote mount that took longest to process.
maxRemoteMountStatPeerName	*string*	Peer name of remote mount that took longest to process.
maxRemoteMountStatTime	*floating point number*	Maximum time spent processing a remote mount in seconds.
maxTimeBetweenActivityLoops	*floating point number*	Maximum time between activity loops in seconds.
mountCompleteCount	*integer*	Number of mounts complete.
mountFlowOutOnNonSchedConn-Count	*integer*	Number of mount FlowOut on non-scheduled connection.
mountRemoteCompleteCount	*integer*	Number of remote mounts complete.
mountRemoteRequestCount	*integer*	Number of remote mounts requested.
mountRequestCount	*integer*	Number of mounts requested.

tacElEntityRefCount	*integer*	TAC entity log reference count.
totalFlowOutCount	*integer*	Total number of FlowOut.

2.77 show agent AGENT memory allocations slab content SLABNAME SLABSIZE partial [...]

blockSize	*integer*	The size of blocks allocated from this allocator.
maxBlocksPerPage	*integer*	The maximum number of blocks we can allocate from a single page.
name	*string*	The name of the allocator, "normal", "transient" or custom name assigned.
perThread	*list of SlabAllocatorContentPerThreadModel* see 2.77.1	Per thread statistics for this allocator.
uniqueName	*string*	Unique name of the allocator.

2.77.1 SlabAllocatorContentPerThreadModel

blocks	*list of SlabAllocatorBlockModel* see 2.77.1.1	Blocks in partially filled pages.
threadName	*string*	The name of the thread the statistics pertains to.

2.77.1.1 SlabAllocatorBlockModel

count	*integer*	Number of blocks seen with this data & onFreeList.
data	*integer*	One pointers worth of data from the block, for blocks in use it is from the start of the block, for free blocks it starts 8 bytes into the block. For blocks not large enough this is always zero.
dataSymbolName	*string*	The symbol name corresponding to key.data.
onFreeList	*boolean*	Is the block free.

2.78 show agent AGENT memory detail

memory	*integer*	Memory usage in kB.
mostCommonObjects	*table(string) = integer*	Counts for the 100 most common objects, keyed by their type.
threads	*integer*	Number of threads in agent.

2.79 show agent AGENT task scheduler [...]

schedulers	*table(string) = TaskScheduler* see 2.79.1	Root schedulers indexed by name.

2.79.1 TaskScheduler

[completedTasks]	*TaskList* see 2.79.1.3	List of completed tasks.
[displayOrder]	*integer*	Rendered text output sorts multiple schedulers in ascending order of (displayOrder, schdulerName).
[generalError] **	*string*	Error encountered running show command. **Since: 4.28.1
[history]	*History* see 2.79.1.1	History of task actions.
isRootScheduler	*boolean*	This is a root scheduler.
maxSchedulerSlip	*floating point number*	Max. scheduler slip, in seconds.
priorityStats	*table(string) = PriorityStats* see 2.79.1.2	Task counts and statistics indexed by priority.
schedulerName	*string*	Name of the scheduler.
subSchedulers	*table(string) = TaskScheduler* see 2.79.1	Nested hierarchical subschedulers indexed by a unique key.
[taskNameRegex] **	*string*	Regular expression used to filter task names. **Since: 4.28.0
[taskNameRegexError] **	*string*	If 'taskNameRegex' is supplied and is not a valid regular expression, this will contain an error message for diagnostic purposes. **Since: 4.28.0
tasks	*TaskList* see 2.79.1.3	List of active tasks.

2.79.1.1 History

history	*list of HistoryEvent* see 2.79.1.1.1	History events.
historyEntryCount	*integer*	Max number of entries in circular history.
historyEventCount **	*integer*	The number of events in circular history. **Since: 4.28.0

2.79.1.1.1 HistoryEvent

[eventId]	*string*	The object associated with the event.
eventType	*string*	Which type of event.
nsSinceBoot	*integer*	Event time, in nanoseconds since system boot.

2.79.1.2 PriorityStats

completedTasks	*integer*	Completed tasks with this priority.
idleTasks	*integer*	Idle tasks with this priority.
runnableTasks	*integer*	Runnable tasks with this priority.
totalRunCount	*integer*	Total run count.

2.79.1.3 TaskList

skipped	*integer*	Skipped tasks with no runs.
tasks	*list of Task* see 2.79.1.3.1	List of tasks.

2.79.1.3.1 Task

[address]	*string*	Memory address of live task.
[completedCount]	*integer*	The number of tasks by this name that have completed.
currentlyRunning	*boolean*	Task is currently running.
firstTaskRunTimestamp **	*integer*	Timestamp of the first time the task ran, in nanoseconds since system boot. If the task has not run it will be zero. **Since: 4.28.0
hasTimer	*boolean*	Task has an associated timer.
lastSchedulingLatency	*floating point number*	Tasks's last scheduling latency, in seconds.
lastTaskRunTimestamp **	*integer*	Timestamp of the most recent time the task ran, in nanoseconds since system boot. If the task as not run it will be zero. **Since: 4.28.0
maxMs	*integer*	Maximum task run time (in ms).
name	*string*	Task name.
onRunQueue	*boolean*	Task is on run queue.
priority	*enum: critical | high | absolute | low | scavenge | normal*	Task priority.
qv200	*integer*	Number of times task used 200% of quantum.
qv500	*integer*	Number of times task used 500% of quantum.
runCount	*integer*	Number of times the Task has run.
[schedulerSuspended] **	*boolean*	Direct or indirect parent scheduler is suspended. **Since: 4.28.0
schedulingLatencyHighWaterMark	*floating point number*	Tasks's scheduling latency high water mark, in seconds.
[subScheduler]	*string*	If the task is a nested hierarchical scheduler, this is identifier in the TaskScheduler.subSchedulers collection.
suspended	*boolean*	Task is suspended.
timerRunning	*boolean*	Task's associated timer is running.
totalRunTime	*floating point number*	Task total run time, in seconds.

[workCount]	*WorkCount* see 2.79.1.3.1.1	Work count.
[workPending]	*WorkPending* see 2.79.1.3.1.2	Pending work.

2.79.1.3.1.1 WorkCount

decayingAvg	*floating point number*	Work count decaying average.
highwater	*integer*	Work count high water mark.
total	*integer*	Total work done.

2.79.1.3.1.2 WorkPending

highwater	*integer*	Pending work high water mark.
pending	*integer*	Pending work.

2.80 show agent cpu [...]

agents	*table(string) = AgentCpuEntry* see 2.80.1	Mapping from Agent name to Agent CPU statistics.

2.80.1 AgentCpuEntry

lastCpuPct	*integer*	Agent current CPU usage (percent).
maxCpuDuration	*integer*	Duration of agent peak CPU usage (seconds).
maxCpuPct	*integer*	Agent peak CPU usage (percent).
pid **	*integer*	Agent process ID. **Since: 4.29.1
threads **	*table(numeric string) = AgentThreadCpuEntry* see 2.80.1.1	Mapping from thread ID to thread CPU statistics. **Since: 4.29.1
totalCpuTime **	*floating point number*	Agent total CPU usage (seconds). **Since: 4.29.1

2.80.1.1 AgentThreadCpuEntry

lastCpuPct	*integer*	Thread current CPU usage (percent).
maxCpuDuration	*integer*	Duration of thread peak CPU usage (seconds).
maxCpuPct	*integer*	Thread peak CPU usage (percent).
name	*string*	Thread name.
totalCpuTime	*floating point number*	Thread total CPU usage (seconds).

2.81 show agent event history [...]

[events]	*list of LogEntryModel* see 2.81.1	List of EventHistory events.
[filters]	*table(string) = string*	Regular expression filters to get a subset of EventHistory.

2.81.1 LogEntryModel

agentName	*string*	Agent name.
category **	*string*	EventHistory category. **Since: 4.21.0
message	*string*	Log message.
messageGroup	*string*	Message group.
timestamp	*floating point number*	Timestamp when the event was logged.

2.82 show agent memory [...]

agents	*table(string) = AgentMemoryEntry* see 2.82.1	Mapping from Agent name to Agent memory statistics.

2.82.1 AgentMemoryEntry

lastVmRss	*integer*	Agent current resident set size (kB).
lastVmSize	*integer*	Agent current virtual memory usage (kB).
maxVmRss	*integer*	Agent peak resident set size (kB).
maxVmSize	*integer*	Agent peak virtual memory usage (kB).

2.83 show agent memory allocations (delta DELTA) [...]

types	*table(string) = AllocTrackTypeDiffModel* see 2.83.1	A mapping of type name to its memory usage information.

2.83.1 AllocTrackTypeDiffModel

currentAllocations	*integer*	Number of objects currently allocated.
deltaAllocations	*integer*	Difference in Number of object allocations.
memoryAllocationOverhead	*integer*	Memory allocation overhead in bytes.
prevAllocations	*integer*	Number of objects allocated when the command was run previously.
size	*integer*	Size of type in bytes.

2.84 show agent memory allocations [...]

types	*table(string) = AllocTrackTypeModel* see 2.84.1	A mapping of type name to its memory usage information.

2.84.1 AllocTrackTypeModel

currentAllocations	*integer*	Number of objects currently allocated.
highestAllocations	*integer*	High watermark of currentAllocations.
memoryAllocationOverhead	*integer*	Memory allocation overhead in bytes.
size	*integer*	Size of type in bytes.
totalAllocations	*integer*	Total number of allocations (calls to new).

2.85 show agent memory allocations heap summary [...]

agents	*table(string) = Mallinfo* see 2.85.1	Heap summary by agent.

2.85.1 Mallinfo

[arena]	*integer*	Non-mmapped space allocated (bytes).
[fordblks]	*integer*	Total free space (bytes).
[fsmblks]	*integer*	Space in freed fastbin blocks (bytes).
[hblkhd]	*integer*	Space allocated in mmapped regions (bytes).
[hblks]	*integer*	Number of mmapped regions.
[keepcost]	*integer*	Top-most, releasable space (bytes).
[ordblks]	*integer*	Number of free chunks.
[smblks]	*integer*	Number of free fastbin blocks.
status	*string*	Heap summary query status (anything other than 'success' indicates an error).
[uordblks]	*integer*	Total allocated space (bytes).
[usmblks]	*integer*	Maximum total allocated space (bytes).

2.86 show agent memory allocations slab [...]

agents	*table(string) = SlabAllocatorAgentModel* see 2.86.1	Per agent statistics, keyed by agent name.
errorAgents	*table(string) = string*	Errors encountered, keyed by agent name.

2.86.1 SlabAllocatorAgentModel

allocatedPages	*integer*	Number of pages allocated to individual allocators or held in per thread free caches.
allocators	*table(string) = SlabAllocatorStatsModel* see 2.86.1.2	Per allocator statistics, keyed by unique name of the allocator.
emptyPages	*integer*	Number of pages that are empty, but dirty.
emptyPagesPerThread	*list of SlabAllocatorEmptyPagesPerThreadModel* see 2.86.1.1	Number of empty pages in the per thread empty page cache.
mmapPages	*integer*	Number of pages of virtual address mmap'd by the slab allocator.
pageSize	*integer*	Size of the internal pages in bytes.
releasedPages	*integer*	Number of pages that were used in the past, but released back to the kernel, these do not consume physical memory.
touchedPages	*integer*	Number of pages touched by the slab allocator.

2.86.1.1 SlabAllocatorEmptyPagesPerThreadModel

emptyPages	*integer*	Number of empty pages.
threadName	*string*	The name of the thread the statistics pertains to.

2.86.1.2 SlabAllocatorStatsModel

blockSize	*integer*	The size of blocks allocated from this allocator.
maxBlocksPerPage	*integer*	The maximum number of blocks we can allocate from a single page.
name	*string*	The name of the allocator, "normal", "transient" or custom name assigned.
perThread	*list of SlabAllocatorStatsPerThreadModel* see 2.86.1.2.1	Per thread statistics for this allocator.

2.86.1.2.1 SlabAllocatorStatsPerThreadModel

blocksFree	*integer*	Number of blocks free.
blocksInUse	*integer*	Number of blocks in use.
fragmentationScore **	*floating point number*	Fragmentation score: pages*free/used. **Since: **currentRelease**

pages	*integer*	Number of pages used by this allocator in this thread.
pagesWithBlocksFreeCount	*table(numeric string) = integer*	Key is number of blocks in a page that is free, value is the number of pages with that number of blocks free.
threadName	*string*	The name of the thread the statistics pertains to.

2.87 show agent memory allocations slab sort ((blocks (in-use | free | percent-free)) | (memory (in-use | free | percent-free | overhead | total | mmap | madvise | dirty)))

see 2.86

2.88 show agent names [...]

agentNames	*list of string*	List of Agent names in Sysdb's launcher config.

2.89 show agent oom scores [...]

agents	*table(string) = AgentOomScoresEntry* see 2.89.1	Mapping from Agent name to Agent OOM scores information.

2.89.1 AgentOomScoresEntry

maxVmSize	*integer*	Agent peak virtual memory size.
oomScore	*integer*	Out-of-memory score of agent: -1000 (least likely to be a victim when system needs to find memory) to 1000.
oomScoreAdj	*integer*	Adjustment up or down (1000 to -1000) to apply to the computed OOM-score of the agent in order to make it more or less likely to become a target/victim of out-of-memory killer).
rssAnon	*integer*	Agent resident anonymous memory size.
vmSize	*integer*	Agent virtual memory size.

2.90 show agent ping [...]

agents	*table(string) = AgentPingEntry* see 2.90.1	Mapping from Agent name to Agent ping statistics.

2.90.1 AgentPingEntry

[lastPingRtt]	*floating point number*	RTT in seconds of the most recent agent ping event.
[lastPingTimestamp]	*integer*	Epoch timestamp in seconds of the most recent agent ping event.
[maxPingRtt]	*floating point number*	Max RTT in seconds observed for agent ping event.
[maxPingTimestamp]	*integer*	Epoch timestamp in seconds when max RTT value was recorded.

2.91 show agent qtrace event tracking [...]

[events]	*list of QtEventTracking-EntryModel* see 2.91.1	List of wall-clock events from qt file.

2.91.1 QtEventTrackingEntryModel

eventData	*string*	Qt wall-clock event data.
eventKey	*string*	Qt wall-clock event message.
eventType	*string*	Qt wall-clock event type.
timestamp	*floating point number*	UTC timestamp when the event occured.

2.92 show agent qtrace size [...]

agents	*table(string) = AgentQtSizeSpecs* see 2.92.1	Map an agent's name to its set of SizeSpecs.

2.92.1 AgentQtSizeSpecs

handles	*table(string) = AgentQtSizeSpec* see 2.92.1.1	Map a qt handle name to its SizeSpec in an agent.

2.92.1.1 AgentQtSizeSpec

sizes	*list of integer*	List of sizes in KB in a quicktrace handle in an agent.

2.93 show agent runnability [...]

agents	*table(string) = AgentRunnability* see 2.93.1	Mapping from agent name to agent runnability status.

2.93.1 AgentRunnability

agentName	*string*	Agent name.
reason	*list of string*	The reasons the agent is or is not runnable.
role	*string*	The redundancy role the agent is expected to run in.
status	*boolean*	True if this agent is runnable.

2.94 show agent uptime [...]

agents	*table(string) = AgentUptimeEntry* see 2.94.1	Mapping from Agent name to Agent restart statistics.

2.94.1 AgentUptimeEntry

agentRunning **	*boolean*	Whether agent is currently running. **Since: 4.28.2
agentStartTime	*integer*	Agent start time in UTC seconds since epoch.
restartCount	*integer*	Number of agent restarts.

2.95 show agent warm-restart [...]

agents	*table(string) = ShowAgentWarmRestartEntry* see 2.95.1	Mapping from Agent name to Agent warm restart configurations.

2.95.1 ShowAgentWarmRestartEntry

allowedCount	*integer*	Number of warm restarts allowed.
windowLength	*floating point number*	Window length in seconds for warm restarts.

2.96 show aliases [...]

aliases	*table(string) = Alias* see 2.96.1	Maps alias names to their corresponding commands.
commandsPath **	*string*	Directory of scripts to be converted to aliases. **Since: 4.18.1
dynamicAliases **	*table(string) = Alias* see 2.96.1	Maps aliases of scripts in commandsPath to their corresponding commands. **Since: 4.18.1
regexAliases **	*list of RegexAlias* see 2.96.2	Regular expression aliases. **Since: 4.24.0

2.96.1 Alias

Could also be an instance of:
RegexAlias 2.96.2

commands	*table(numeric string) = string*	Map a sequence number to a command.

2.96.2 RegexAlias

commands	*table(numeric string) = string*	Map a sequence number to a command.
regex	*string*	A Python-compatible regular expression.

2.97 show application traffic recognition [...]

applications	*table(string) = ApplicationModel* see 2.97.1	Applications configured on the switch, keyed by application name.

2.97.1 ApplicationModel

classifier	*enum: customDpi | ipv4 | dpi*	The classifier used for the application.
services	*table(string) = ServiceModel* see 2.97.1.1	Services available for the application, keyed by service name.

2.97.1.1 ServiceModel

category	*string*	Category assigned to the application service.

2.98 show application traffic recognition counters [...]

applications	*table(string) = ApplicationCountersModel* see 2.98.1	Applications configured on the switch, keyed by application name.

2.98.1 ApplicationCountersModel

appId	*integer*	Application identifier.
services	*table(string) = ServiceVrfsModel* see 2.98.1.1	Services available for the application, keyed by service name.

2.98.1.1 ServiceVrfsModel

serviceId	*integer*	Service identifier.
vrfs	*table(string) = Vrf-ApplicationCountersModel* see 2.98.1.1.1	Application service data keyed by VRF name.

2.98.1.1.1 VrfApplicationCountersModel

categories	*table(string) = integer*	A mapping of category names to ids.
counters	*ApplicationDataModel* see 2.98.1.1.1.1	Counters for an application service.
profiles	*table(string) = integer*	A mapping of profile names to ids.
vrfId	*integer*	VRF identifier.

2.98.1.1.1.1 ApplicationDataModel

classifierPackets	*integer*	Number of packets processed by the classifier.
totalBytes	*integer*	Number of bytes received for an application service.
totalFlows	*integer*	Number of flows received for an application service.
totalPackets	*integer*	Number of packets received for an application service.

2.99 show arp agent (ipv4 | ipv6) cache capacity [...]

capacityKinds	*table(string) = CapacityKinds* see 2.99.1	Mapping from cache entry kind to neighbor cache capacity kind.

2.99.1 CapacityKinds

interfaces	*table(Interface name) = Capacity* see 2.99.1.1	Mapping from interface to neighbor cache capacity.

2.99.1.1 Capacity

cacheEntries	*list of IP address*	List of IP addresses of the neighbor cache entries.
capacity	*integer*	Capacity of the neighbor cache.
count	*integer*	Count of the neighbor cache entries.

2.100 show arp agent (ipv4 | ipv6) input [...]

entryMap	*table(string) = ArpInputEntryList* see 2.100.1	Mapping of VRF name to list of ARP entries.

2.100.1 ArpInputEntryList

entrys	*list of ArpInputEntry* see 2.100.1.1	ARP input entry list.

2.100.1.1 ArpInputEntry

entryType	*enum: dynamic | static*	Type of entry (Static/Dyn).
ethAddr	*string*	MAC address.
genId	*string*	Gen ID of dynamic entries.
intfId	*interface name*	Name of the interface.
ipAddr	*IP address*	Ip or Ipv6 address.
refreshTime	*string*	Quick Refresh time.
source	*string*	Source of the ARP entries.

2.101 show arp agent (ipv4 | ipv6) output [...]

see 2.100

2.102 show arp monitor [...]

entryList	*list of MonitoredNexthopEntry* see 2.102.1	Monitored nexthop entry.
totalEntries	*integer*	Total number of monitored nexthop entries.
vrfList	*list of MonitoredNexthopVrfSummary* see 2.102.2	Monitored nexthop entry count.

2.102.1 MonitoredNexthopEntry

intf	*interface name*	Name of the interface.
ipAddr	*string*	Nexthop Ip or Ipv6 address.

2.102.2 MonitoredNexthopVrfSummary

entries	*integer*	Total number of monitored nexthop entries for the Vrf.
vrfname	*string*	Name of the Vrf.

2.103 show arp monitor summary

see 2.102

2.104 show arp | (ipv6 neighbors) bridged [...]

see 2.40

2.105 show banner login

`loginBanner`	*string*	Login banner for the switch.

2.106 show banner motd

`motd`	*string*	Message of the day for the switch.

2.107 show bfd ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.108 show bfd (neighbors | peers) [...]

revision 4

vrfs	*table(string) = BfdNeighborVrf* see 2.108.1	A mapping between vrf and peer stats.

2.108.1 BfdNeighborVrf

ipv4InitiatorNeighbors	*table(IPv4 or IPv6 address) = SbfdInitiatorTunnel* see 2.108.1.2	A mapping between IPv4 peer address and tunnel ID.
ipv4Neighbors	*table(IPv4 or IPv6 address) = BfdNeighborInterface* see 2.108.1.1	A mapping between IPv4 peer address and interface.
ipv4ReflectorNeighbors	*table(IPv4 or IPv6 address) = SbfdReflector* see 2.108.1.3	A mapping between IPv4 peer address and remote discriminators.
ipv6InitiatorNeighbors	*table(IPv4 or IPv6 address) = SbfdInitiatorTunnel* see 2.108.1.2	A mapping between IPv6 peer address and tunnel ID.
ipv6Neighbors	*table(IPv4 or IPv6 address) = BfdNeighborInterface* see 2.108.1.1	A mapping between IPv6 peer address and interface.
ipv6ReflectorNeighbors	*table(IPv4 or IPv6 address) = SbfdReflector* see 2.108.1.3	A mapping between IPv6 peer address and remote discriminators.

2.108.1.1 BfdNeighborInterface

peers	*table(Interface name) = BfdNeighborIntfPeer* see 2.108.1.1.1	A mapping between interface and peer type.

2.108.1.1.1 BfdNeighborIntfPeer

types	*table(string) = BfdNeighborSrcAddr* see 2.108.1.1.1.1	A mapping between peer type and source address.

2.108.1.1.1.1 BfdNeighborSrcAddr

peerStats	*table(IPv4 or IPv6 address) = PeerStats* see 2.108.1.1.1.2	A mapping between source address and peer stats.

2.108.1.1.1.2 PeerStats

authProfileName	*string*	Configured authentication profile.

authType	*enum: None* | *Meticulous Sha1* | *Keyed Md5* | *Password* | *Meticulous Md5* | *Keyed Sha1*	Authentication mode.
[destPort]	*integer*	Destination Port of SBFD Reflector.
[echoRxStats]	*RxStats* see 2.108.1.1.1.7	Echo Rx statistics for peer.
[echoTxStats]	*TxStats* see 2.108.1.1.1.8	Echo Tx statistics for peer.
[kernelIfIndex]	*integer*	Kernel interface.
l3intf	*interface name*	Local egress interface.
lastDiag	*enum: diagEchoFail* | *diagConcatPathDown* | *diagCtrlTimeout* | *diagAdminDown* | *diagNeighDown* | *diagRevConcatPathDown* | *diagForwardingReset* | *diagNone* | *diagPathDown*	Last diagnostic for reason of change to Down state.
lastDown	*floating point number*	Last time peer session went down in seconds, 0.0 if never down.
lastUp	*floating point number*	Last time peer session came up in seconds, 0.0 if never up.
localDisc	*integer*	Local discriminator of BFD session.
[peerStatsDetail]	*PeerStatsDetail* see 2.108.1.1.1.3	Detailed peer stats.
proxy	*boolean*	S-BFD proxy Enabled.
remoteDisc	*integer*	Remote discriminator BFD session.
[rttStats]	*RttStats* see 2.108.1.1.1.6	RTT statisics for peer.
[rxStats]	*RxStats* see 2.108.1.1.1.7	Rx statistics for peer.
sessType	*enum: sessionTypeMicroLegacy* | *sessionTypeLagLegacy* | *sessionTypeMultihop echo* | *sessionTypeSbfdReflector* | *sessionTypeSbfdInitiator* | *sessionTypeLagRfc7130 echo* | *sessionTypeMultihop* | *sessionTypeVxlanTunnel* | *sessionTypeMicroLegacy echo* | *sessionTypeNormal* | *sessionTypeMicroRfc7130* | *sessionTypeVxlanTunnel echo* | *sessionTypeLagRfc7130* | *sessionTypeMicroRfc7130 echo* | *sessionTypeLagLegacy echo* | *sessionTypeNormal echo*	BFD session type.
status	*enum: down* | *adminDown* | *up* | *init*	State of the peer.

[tunnelId]	*integer*	TunnelID of SBFD Initiator.
[tunnelIdx]	*integer*	Tunnel index of SBFD Initiator.
[tunnelType]	*enum: srTeSegmentListTunnel | eosSdkSbfdSessionTunnel*	Tunnel type of SBFD Initiator.
[txStats]	*TxStats* see 2.108.1.1.1.8	Tx statistics for peer.

2.108.1.1.1.3 PeerStatsDetail

apps	*list of string*	Registered protocols.
detectMult	*integer*	Number of consecutive BFD packets missed before failure.
detectTime	*integer*	Detection timeout in microseconds.
echoActive	*boolean*	Active echo session with both tx and rx session.
echoDisc	*integer*	Discriminator of BFD Echo session.
echoOn	*boolean*	Echo turned on.
hwAcceleratedStates	*table(string) = boolean*	Mapping of session type to hw acceleration state.
lastDemand	*boolean*	Last received Peer demand bit.
lastDetectMult	*integer*	Last received Peer detection multiplier.
lastDiag	*integer*	Last received Peer diagnostic code between values 0-8.
lastEchoRxIntv	*integer*	Last received Echo Rx interval in microseconds.
lastFinal	*boolean*	Last received Peer final bit.
lastLength	*integer*	Last received packet length.
lastMinRxIntv	*integer*	Last received minimum Rx interval in microseconds.
lastMinTxIntv	*integer*	Last received minimum Tx interval in microseconds.
lastMyDiscU32	*integer*	Last received Peer discriminator.
lastPoll	*boolean*	Last received Peer poll bit.
lastState	*enum: down | adminDown | up | init*	Last received Peer BFD session state.
lastVersion	*integer*	Last received Peer BFD version.
lastYourDiscU32	*integer*	Last received local discriminator.
localIpAddr	*IP address*	Local IP.
operEchoTxRxInterval	*integer*	Operational rate in microseconds at which Bfd echo packets are sent and received.
operRxInterval	*integer*	Operational rate in microseconds at which BFD control pckets are received.
operTxInterval	*integer*	Operational rate in microseconds at which BFD control packets are sent.
remoteDetectMult	*integer*	Receive multiplier of peer.
remoteMinRxInterval	*integer*	Minimum rate in microseconds at which BFD packets can be received by the peer.
role	*enum: passive | active*	BFD role.
segmentType	*enum: invalid | mplsLabel | ipNexthop*	Segment type.

segments	*list of integer*	Proxy segment list.
[tunnelInfo]	*TunnelInfo* see 2.108.1.1.1.4	Tunnel information of tunnelId.

2.108.1.1.1.4 TunnelInfo

ipDscp	*integer*	IP DSCP value.
mplsExp	*integer*	MPLS EXP value.
mplsLabels	*list of integer*	MPLS label stack.
[sbfdReturnPathLabelStack]	*list of integer*	MPLS label stack of SBFD return path.
[via]	*Via* see 2.108.1.1.1.5	Via next hop.

2.108.1.1.1.5 Via

interface	*interface name*	Egress L3 interface of the next hop.
nextHop	*IP address*	Next hop IP address.

2.108.1.1.1.6 RttStats

avgRtt	*integer*	Average RTT in microseconds.
lastRtt	*integer*	Last RTT in microseconds.
maxRtt	*integer*	Maximum RTT in microseconds.
minRtt	*integer*	Minimum RTT in microseconds.

2.108.1.1.1.7 RxStats

[avgPeriod]	*integer*	Average time between received packets in microseconds.
[lastRxState]	*enum: down | adminDown | up | init*	Last Rx state of the peer.
[lastRxStateChg]	*floating point number*	Time of last Rx packet of state change received in seconds.
[lastRxTime]	*floating point number*	Time of last successfully authenticated and validated packet received in seconds.
[maxPeriod]	*integer*	Maximum time between received packets in microseconds.
[minPeriod]	*integer*	Minimum time between received packets in microseconds.
numReceived	*integer*	Number of packets received.

2.108.1.1.1.8 TxStats

[avgLateness]	*integer*	Average number of microseconds messages sent after its configured period.
[lastTxTime]	*floating point number*	Time of last successfully transmitted packet in seconds.
[maxLateness]	*integer*	Max number of microseconds a message sent after its configured period.

[minLateness]	*integer*	Minimum number of microseconds a message sent after its configured period.
numSent	*integer*	Number of packets sent since last Tx config change.
[sentAfter3p]	*integer*	Number of packets sent later than three periods of last packet.
[sentWithin1p]	*integer*	Number of packets sent within configured period since last packet.
[sentWithin2p]	*integer*	Number of packets sent later than one period but before two periods of last packet.
[sentWithin3p]	*integer*	Number of packets sent later than two periods but before three periods of last packet.

2.108.1.2 SbfdInitiatorTunnel

peerStats	*table(numeric string) = PeerStats* see 2.108.1.1.1.2	A mapping between tunnel ID and peer stats.

2.108.1.3 SbfdReflector

peerStats	*table(numeric string) = PeerStats* see 2.108.1.1.1.2	A mapping between remote discriminator and peer stats.

2.109 show bfd (neighbors | peers) debug [...]

downSessions	*table(string) = integer*	Mapping of diag to number of down sessions.
echoRxIntervals	*table(string) = string*	Mapping of peer to echo rx interval.
issues	*table(string) = Issue* see 2.109.1	Mapping of issue type to issue detail.
rxIntervals	*table(string) = string*	Mapping of peer to rx interval.
slowEchoTxs	*table(string) = integer*	Mapping of peer to number of slow echo tx.
slowTxs	*table(string) = integer*	Mapping of peer to number of slow tx.

2.109.1 Issue

issueList	*list of string*	List of issues.

2.110 show bfd (neighbors | peers) history [...]

peerSnapshots	*list of PeerSnapshot* see 2.110.1	List of BFD peer history.

2.110.1 PeerSnapshot

authProfileName **	*string*	Configured authentication profile. **Since: 4.22.0
authType **	*enum: None | Meticulous Sha1 | Keyed Md5 | Password | Meticulous Md5 | Keyed Sha1*	Authentication mode. **Since: 4.22.0
[avgLateness] **	*integer*	Average number of microseconds messages sent after its configured period. **Since: 4.29.2
[avgPeriod] **	*integer*	Average time between received packets in microseconds. **Since: 4.29.2
detectTime	*integer*	Detection timeout in microseconds.
[echoAvgLateness] **	*integer*	Average number of microseconds messages sent after its configured period for echo. **Since: 4.29.2
[echoAvgPeriod] **	*integer*	Average time between received packets in microsecondsfor echo. **Since: 4.29.2
echoDetectTime **	*integer*	Detection timeout in microseconds for echo. **Since: 4.22.0
[echoLastRxTime] **	*floating point number*	Time of last successfully authenticated and validated packet received in seconds for echo. **Since: 4.29.2
[echoLastTxTime] **	*floating point number*	Time of last successfully transmitted packet in seconds for echo. **Since: 4.29.2
[echoMaxLateness] **	*integer*	Max number of microseconds a message sent after its configured period for echo. **Since: 4.29.2
[echoMaxPeriod] **	*integer*	Maximum time between received packets in microsecondsfor echo. **Since: 4.29.2
[echoMinLateness] **	*integer*	Minimum number of microseconds a message sent after its configured period for echo. **Since: 4.29.2
[echoMinPeriod] **	*integer*	Minimum time between received packets in microseconds for echo. **Since: 4.29.2
echoOn **	*boolean*	Echo turned on. **Since: 4.22.0

`echoRxCount` **	*integer*	Number of echo packets received. **Since: 4.22.0
`echoRxInterval` **	*integer*	Rate in microseconds at which BFD echo packets are received. **Since: 4.22.0
`echoSentAfter3p` **	*integer*	Number of echo packets sent later than three periods of last echo packet. **Since: 4.22.0
`echoSentWithin1p` **	*integer*	Number of echo packets sent within configured period since last echo packets. **Since: 4.22.0
`echoSentWithin2p` **	*integer*	Number of echo packets sent later than one period but before two periods of last echo packet. **Since: 4.22.0
`echoSentWithin3p` **	*integer*	Number of echo packets sent later than two periods but before three periods of last echo packet. **Since: 4.22.0
`echoTxCount` **	*integer*	Number of echo packets sent. **Since: 4.22.0
`echoTxInterval` **	*integer*	Rate in microseconds at which BFD echo packets are sent. **Since: 4.22.0
`event`	*enum: sessionDown* | *sessionDelete* | *sessionNoEvent* | *sessionPeerDown* | *sessionPeerAdminDown* | *sessionAdminDown*	BFD session event.
`eventTime`	*floating point number*	Time of BFD session event in seconds.
`intf`	*interface name*	Local egress interface.
`ip`	*IP address*	IP address of peer.
`lastDiag`	*enum: diagEchoFail* | *diagConcatPathDown* | *diagCtrlTimeout* | *diagAdminDown* | *diagNeighDown* | *diagRevConcatPathDown* | *diagForwardingReset* | *diagNone* | *diagPathDown*	Last diagnostic for reason of change to Down state.
`lastDown`	*floating point number*	Last time peer session went down in seconds, 0.0 if never down.
`[lastRxTime]` **	*floating point number*	Time of last successfully authenticated and validated packet received in seconds. **Since: 4.29.2
`[lastTxTime]` **	*floating point number*	Time of last successfully transmitted packet in seconds. **Since: 4.29.2
`lastUp`	*floating point number*	Last time peer session went up in seconds, 0.0 if never up.
`localDisc` **	*integer*	Local discriminator of BFD session. **Since: 4.29.2

localIpAddr **	*IP address*	Local IP. **Since: 4.29.2
[maxLateness] **	*integer*	Max number of microseconds a message sent after its configured period. **Since: 4.29.2
[maxPeriod] **	*integer*	Maximum time between received packets in microseconds. **Since: 4.29.2
[minLateness] **	*integer*	Minimum number of microseconds a message sent after its configured period. **Since: 4.29.2
[minPeriod] **	*integer*	Minimum time between received packets in microseconds. **Since: 4.29.2
prevState **	*enum: down* | *adminDown* | *up* | *init*	previous state. **Since: 4.29.0
remoteDisc **	*integer*	Remote discriminator BFD session. **Since: 4.29.2
rxCount	*integer*	Number of packets received.
rxInterval	*integer*	Rate in microseconds at which BFD control packets are received.
sentAfter3p	*integer*	Number of packets sent later than three periods of last packet.
sentWithin1p	*integer*	Number of packets sent within configured period since last packet.
sentWithin2p	*integer*	Number of packets sent later than one period but before two periods of last packet.
sentWithin3p	*integer*	Number of packets sent later than two periods but before three periods of last packet.
status	*enum: down* | *adminDown* | *up* | *init*	State of peer session.
[tunnelId] **	*integer*	TunnelID of SBFD Initiator. **Since: 4.24.0
txCount	*integer*	Number of packets sent.
txInterval	*integer*	Rate in microseconds at which BFD control packets are sent.
vrf	*string*	Vrf that peer is configured in.

2.111 show bfd (neighbors | peers) summary [...]

`adminDown` **	*boolean*	BFD has been shut down administratively. **Since: 4.23.0
`initiatorMinTx` **	*integer*	Operational rate in microseconds at which SBFD control packets are sent on initiator. **Since: 4.24.2
`initiatorMult` **	*integer*	Number of consecutive SBFD packets missed before failure. **Since: 4.24.2
`[localIntf]` **	*interface name*	SBFD local interface. **Since: 4.24.1
`localIntfIpAdd` **	*IP address*	Local interface IP address. **Since: 4.24.2
`mhMinRx` **	*integer*	Operational rate in microseconds at which BFD control pckets are received on multi-hop BFD session. **Since: 4.23.0
`mhMinTx` **	*integer*	Operational rate in microseconds at which BFD control packets are sent on multihop BFD session. **Since: 4.23.0
`mhMult` **	*integer*	Number of consecutive BFD packets missed before failure onmultihop BFD session. **Since: 4.23.0
`minRx` **	*integer*	Operational rate in microseconds at which BFD control pckets are received. **Since: 4.23.0
`minTx` **	*integer*	Operational rate in microseconds at which BFD control packets are sent. **Since: 4.23.0
`mult` **	*integer*	Number of consecutive BFD packets missed before failure. **Since: 4.23.0
`reflectorLocalDisc` **	*integer*	Local discriminator of SBFD session. **Since: 4.24.2
`reflectorMinRx` **	*integer*	Operational rate in microseconds at which SBFD control pckets are received on reflector. **Since: 4.24.2
`sessionStatsInterval` **	*integer*	Interval in seconds between unsolicited per-session statistics from BFD acclerators. **Since: 4.25.2
`sessions`	*table(string) = SessionCountByType* see 2.111.1	Mapping of addressing to session count by type.
`slowTimer` **	*integer*	Operation rate in microseconds at which BFD control packets are received on async sessions when echo is enabled. **Since: 4.23.0

2.111.1 SessionCountByType

types	*table(string) = SessionCountByState* see 2.111.1.1	Mapping of session type to session count by state.

2.111.1.1 SessionCountByState

states	*table(string) = SessionCount* see 2.111.1.1.1	Mapping of session state to number of sessions.

2.111.1.1.1 SessionCount

async	*integer*	Number of async sessions
echo	*integer*	Number of echo sessions.

2.112 show bfd hardware acceleration

reasons **	*list of string*	List of reasons why hardware acceleration is not running. **Since: 4.23.0
running	*boolean*	Hardware acceleration is running.
supported	*boolean*	Hardware acceleration is supported by platform.

2.113 show bfd hardware utilization [...]

resources	*table(string) = BfdHwResource* see 2.113.1	Hardware resource information keyed by the resource's name.

2.113.1 BfdHwResource

maxSessions	*integer*	Maximum number of sessions configured of the chip.
[numSessions]	*integer*	Number of sessions configured on a chip.
[sessions]	*list of BfdHwResourceSession* see 2.113.1.1	List of sessions configured on the chip.

2.113.1.1 BfdHwResourceSession

intf	*interface name*	Local egress interface.
ip	*IP address*	IP address of peer.
localDisc	*integer*	Local discriminator of BFD session.

sessType	*enum:* *sessionTypeMicroLegacy* | *sessionTypeLagLegacy* | *sessionTypeMultihop echo* | *sessionTypeSbfdReflector* | *sessionTypeSbfdInitiator* | *sessionTypeLagRfc7130 echo* | *sessionTypeMultihop* | *sessionTypeVxlanTunnel* | *sessionTypeMicroLegacy echo* | *sessionTypeNormal* | *sessionTypeMicroRfc7130* | *sessionTypeVxlanTunnel echo* | *sessionTypeLagRfc7130* | *sessionTypeMicroRfc7130 echo* | *sessionTypeLagLegacy echo* | *sessionTypeNormal echo*	BFD session type.
vrf	*string*	VRF that peer is configured in.

2.114 show bfd peers protocol vxlan mlag primary

interfaces	*table(Interface name) = VxlanVtepBfdStatus* see 2.114.1	Mapping from VXLAN interface to its VTEP addresses.

2.114.1 VxlanVtepBfdStatus

vteps	*table(IPv4 or IPv6 address) = string*	A mapping of VXLAN VTEP to its BFD status.

2.115 show bfd rbfd-stats

netlinkFailureFail	*integer*	Number of times session failure sent but broadcast failed.
netlinkPacketChangeFail	*integer*	Number of times packet change sent but broadcast failed.
numInvalidSessions	*integer*	Number of packets without session in Rx.
numPassupDisc0	*integer*	Number of packets passed up with discriminator zero.
pskbExpandFail	*integer*	Number of times pskb needed to be expanded but failed for packets without a session.
rfcDrops	*integer*	Number of packets dropped on receive before processing.
rxPacketMissing	*integer*	Number of times packet missing upon Rx trigger.

2.116 show bfd sbfd-stats reflectors

rxDemandUnset	*integer*	Number of packets with demand mode unset on receive.
rxDrop	*integer*	Number of packets dropped on receive.
rxHashHit	*integer*	Number of session cache lookup hit.
rxHashMiss	*integer*	Number of session cache lookup miss.
rxHashTrim	*integer*	Number of session cache lookup trim.
rxInvalidBfdState	*integer*	Number of packets with invalid bfd state on receive.
rxInvalidLen	*integer*	Number of packets with invalid length on receive.
rxInvalidMulti	*integer*	Number of packets with invalid multiple detect on receive.
rxInvalidMyDisc	*integer*	Number of packets with invalid my discriminator on receive.
rxInvalidPktFormat	*integer*	Number of packets with invalid packet format on receive.
rxInvalidSrcPort	*integer*	Number of packets with invalid source port on receive.
rxInvalidTlv **	*integer*	Number of packets with invalid TLVs on receive. **Since: 4.28.2
rxInvalidVer	*integer*	Number of packets with invalid version on receive.
rxTruncated	*integer*	Number of packets truncated on receive.
rxUnsupported	*integer*	Number of packets unsupported on receive.
rxValid	*integer*	Number of valid packets received.
rxYourDiscUnmatch	*integer*	Number of packets with unmatched your discriminator on receive.
txInvalidLen	*integer*	Number of packets with invalid length.
txSend	*integer*	Number of packets sent.
txSendToFail	*integer*	Number of packets failed to send.

2.117 show bgp ((vpn-ipv4 | vpn-ipv6)) summary

revision 3

see 2.46

2.118 show bgp (AFI SAFI) ((IP | IP_PREFIX)) [...]

revision 3

see 2.17

2.119 show bgp (AFI SAFI) [...]

revision 3

see 2.17

2.120 show bgp (AFI SAFI) summary [...]

revision 3

see 2.46

2.121 show bgp (AFI SAFI) summary lldp [...]

revision 3

see 2.46

2.122 show bgp AFI unicast field-set FSNAME [...]

revision 3

see 2.17

2.123 show bgp access-list [...]

see 2.53

2.124 show bgp auto-aggregation ipv4 [...]

`vrfs`	*table(string) = AutoAggregation* see 2.124.1	Per VRF auto aggregation summary.

2.124.1 AutoAggregation

`domains`	*list of Domain* see 2.124.1.1	Auto aggregation domains.

2.124.1.1 Domain

`address`	*IP address*	Domain address.
`contiguousPrefixBlocks`	*list of PrefixBlock* see 2.124.1.1.1	List of contiguous prefix blocks.
`maskLength`	*integer*	Domain mask length.

2.124.1.1.1 PrefixBlock

firstAddress	*IP address*	Block first address.
lastAddress	*IP address*	Block last address.
prefixes	*list of Prefix* see 2.124.1.1.1.1	Prefixes that are active in the RIB and eligible for auto aggregation.

2.124.1.1.1.1 Prefix

address	*IP address*	Prefix address.
autoAggregate	*boolean*	Whether this prefix is an auto aggregate.
contributors	*list of Contributor* see 2.124.1.1.1.2	Auto aggregate contributors.
maskLength	*integer*	Prefix mask length.
[notAggregatedReason]	*noComplement*	Reason this prefix is not auto aggregated.

2.124.1.1.1.2 Contributor

address	*IP address*	Contributor address.
maskLength	*integer*	Contributor mask length.

2.125 show bgp auto-aggregation ipv6 [...]

see 2.124

2.126 show bgp convergence [...]

vrfs	*table(string) = ShowBgpConvergenceVrfModel* see 2.126.1	Per VRF BGP convergence information.

2.126.1 ShowBgpConvergenceVrfModel

config	*ShowBgpConvergenceVrfConfigModel* see 2.126.1.1	Convergence configration for VRF.
peersNotConverged	*table(IPv4 or IPv6 address) = string*	Current state of peers that still haven't converged in VRF, keyed by peer address.
status	*ShowBgpConvergenceVrfStatusModel* see 2.126.1.2	Current convergence status for VRF.

2.126.1.1 ShowBgpConvergenceVrfConfigModel

afiSafiWaitForConvergence	*table(string) = boolean*	Wait for convergence enabled for each AFI/SAFI.
convergenceTimeout	*integer*	Convergence timeout in seconds.
[convergenceUpdateSyncEnabled]	*boolean*	Convergence based update synchronisation is enabled.
slowPeerTimeout	*integer*	Slow peer timeout in seconds.

2.126.1.2 ShowBgpConvergenceVrfStatusModel

[afiSafiConvergenceStates]	*table(string) = string*	Convergence state per AFI/SAFI type.
[allIgpConverged]	*boolean*	All IGP protocols have converged.
convergenceState	*enum: notInitialized | convergenceIdle | converged | convergenceActive | timeout*	BGP convergence states.
[convergenceTimeout]	*floating point number*	Convergence timeout in seconds.
[convergenceTimer]	*boolean*	Convergence timer is running.
[convergenceTimerExpires]	*floating point number*	Time since epoch when timer expires.
[dynamicPeerGroups]	*boolean*	Dynamic peering is configured.
[expectedPeersUp]	*boolean*	Expected peers are up.
[firstPeerUp]	*boolean*	First peer is up.
[firstPeerUpTime]	*floating point number*	Timestamp since epoch when first peer came up.
[lastEventTime]	*floating point number*	Timestamp of last event since epoch.
[outstandingEors]	*integer*	Number of outstanding EORs.
[outstandingKeepalives]	*integer*	Number of outstanding keepalives.
peers	*ShowBgpConvergenceVrfPeerStatsModel* see 2.126.1.2.1	Convergence statistics.
[previouslyEstablishedPeers]	*integer*	Number of previously established peers.
[slowPeerTimeout]	*floating point number*	Slow peer timeout in seconds.

[timeToConverge]	*floating point number*	Seconds taken to converge.

2.126.1.2.1 ShowBgpConvergenceVrfPeerStatsModel

disabledPeers	*integer*	Number of disabled peers.
establishedPeers	*integer*	Number of established peers.
pendingPeers	*integer*	Number of pending peers.
totalPeers	*integer*	Number of total peers.

2.127 show bgp debug policy export ((((vrf-route filter-rcf ipv4 SAFI) | (vpn-ipv4)) PREFIX) | (((vrf-route filter-rcf ipv6 SAFI) | (vpn-ipv6)) PREFIX6)) [...]

revision 2

evalResults	*list of PolicyEvalResult* see 2.127.1	Evaluation result for each NLRI.
[rcfCodeUnits]	*table(string) = RcfCodeUnitText* see 2.127.2	Reference to the text of RCF functions where evaluation occurred in each code unit, keyed by code unit name.

2.127.1 PolicyEvalResult

[messages]	*list of string*	Message describing why policy evaluation was not returned.
nlri	*NlriModel* see 2.127.1.1	NLRI of the evaluated path.
pointOfApplication	*PolicyEvalPointOf-Application* see 2.127.1.2	Point of application evaluated.
[rcfEval]	*RcfDebugEvaluation* see 2.127.1.4	Debug information for an RCF evaluation.
[routeMapEval]	*PolicyEvalRouteMap* see 2.127.1.3	Route map evaluated.

2.127.1.1 NlriModel

[flowspecRule] **	*string*	Bgp flowspec rule. **Since: 4.30.1
[ipPrefix]	*IP address prefix*	IPv4 or IPv6 prefix.
nlriType	*enum: ipv4Flowspec | evpnType5Ipv4 | mplsVpnIpv6Unicast | evpnType5Ipv6 | ipv6Flowspec | ipv6Unicast | ipv4Unicast | mplsVpnIpv4Unicast*	Type of BGP NLRI.
[pathId] **	*integer*	Path ID of the route. **Since: 4.30.1
[rd] **	*string*	Route distinguisher. **Since: 4.29.2

2.127.1.2 PolicyEvalPointOfApplication

name	*enum: inbound | export | import | redistribute | outbound*	Name of the point of application.
[peerAddr]	*IP address*	Peer address of application.
[redistributionSource]	*string*	Source of redistributed routes.
[vpnAfiSafi] **	*enum: evpn | ipv4MplsVpn | ipv6MplsVpn*	AfiSafi of VPN routes. **Since: 4.30.0

2.127.1.3 PolicyEvalRouteMap

[notFound]	*boolean*	Route map is not configured.
[routeMapCycleDetected]	*boolean*	Route map is part of a cycle.
routeMapName	*string*	Name of the route map.
routeMapResult	*PolicyEvalRouteMapResult* see 2.127.1.3.1	Result of the route map.
[sequences]	*table(numeric string) = PolicyEvalSequence* see 2.127.1.3.2	Evaluated sequences

2.127.1.3.1 PolicyEvalRouteMapResult

[permittingSeqnos]	*list of integer*	Sequences resulting in permit.
result	*enum: permit | noMatchingSequenceDeny | explicitDeny*	Route map result.

2.127.1.3.2 PolicyEvalSequence

[continueNextSequence]	*boolean*	Continue to next sequence.
[continueSeqno]	*integer*	Continue to sequence number.
[descriptions]	*list of string*	Route map sequence description.
filterType	*enum: permit | deny*	Filter type.
[matcherResults]	*list of MatchRules* see 2.127.1.3.2.1	Evaluated matcher results.
[missingContinueTarget]	*boolean*	Sequence contains a continue statement to a target which is not configured.
sequenceResult	*PolicyEvalSequenceResult* see 2.127.1.3.2.7	Sequence result.
[setRules]	*SetRules* see 2.127.1.3.2.8	Applied set rules.
[subRouteMap]	*PolicyEvalRouteMap* see 2.127.1.3	Sub route map evaluation.
[subRouteMapInvertResult]	*boolean*	Invert result of the sub route map.

2.127.1.3.2.1 MatchRules

[accessList]	*string*	IPv4 address filtered by access control list.
[aggregation]	*AggregateContributor* see 2.127.1.3.2.2	Match contributors to BGP aggregates.
[aggregationInvertResult]	*boolean*	Invert match contributors to BGP aggregates.
[asNumber]	*string*	BGP autonomous system number.
[asPathLength]	*AsPathLengthRange* see 2.127.1.3.2.3	Match on as-path length.
[asPathLengthInvertResult]	*boolean*	Invert of as-path length match.
[asPathList]	*string*	BGP autonomous system path access list.
[community]	*string*	BGP community.
[communityExactMatch]	*boolean*	Exact match of communities.

[communityInstances]	*CommInstancesRange* see 2.127.1.3.2.4	Match on the number of community instances.
[communityInstancesInvertResult]	*boolean*	Invert match community instances result.
[communityInvertResult]	*boolean*	Invert match community result.
[communityOrResults] **	*boolean*	Or results of community lists. **Since: 4.23.1
[debugInfo] **	*DebugInfo* see 2.127.1.3.2.5	Debugging information for match rules. **Since: 4.22.1
[extCommunity]	*string*	BGP extended community.
[extCommunityInvertResult]	*boolean*	Invert match extcommunity result.
[interface]	*interface name*	Interface name.
[ipAddrDynamicPrefixList]	*string*	Ip address dynamic prefix list name.
[ipv6AddrDynamicPrefixList]	*string*	Ipv6 address dynamic prefix list name.
[ipv6NextHop]	*IPv6 address*	IPv6 next hop filtered by IPv6 address.
[ipv6NextHopPrefixList]	*string*	IPv6 next hop filtered by IPv6 prefix list.
[ipv6PrefixList]	*string*	IPv6 address filtered by IPv6 prefix list.
[ipv6ResolvedNextHopPrefixList]	*string*	IPv6 resolved nexthop filtered by IPv6 prefix list.
[isisInstance] **	*integer*	IS-IS route instance. **Since: 4.24.2
[isisLevel]	*enum: level-2 | level-1*	IS-IS route level.
[largeCommunity]	*string*	BGP large community.
[largeCommunityInvertResult]	*boolean*	Invert match large community result.
[localPref]	*integer*	Bgp local preference metric.
[nextHop]	*IPv4 address*	IPv4 next hop filtered by IP address.
[nextHopPrefixList]	*string*	IPv4 next hop filtered by IP prefix list.
[originAs] **	*string*	Route origin AS number. **Since: 4.23.1
[originAsValidity] **	*enum: valid | notFound | invalid*	AS Origin validation state. **Since: 4.23.2
[originAsValidityInvertResult] **	*boolean*	Invert match AS Origin validation state. **Since: 4.24.0
[ospfInstance] **	*integer*	OSPF route instance. **Since: 4.25.0
[ospfMetricType]	*enum: type-2 | type-1*	OSPF external LSA metric type.
[prefixList]	*string*	IPv4 address filtered by IP prefix list.
[protocol]	*enum: ospf | dynamic | connected | static | bgp | ospfv3 | rip | isis | attached-host*	Routing protocol of route's source.
[resolvedNextHopPrefixList]	*string*	IPv4 resolved nexthop filtered by IP prefix list.
[routeMetric]	*integer*	Route metric.
[routeTag]	*integer*	Route tag.
[routeType]	*enum: confederation-external | internal | local | vpn | external*	Route type.
[routerIdPrefixList]	*string*	Router ID filtered by prefix list.

2.127.1.3.2.2 AggregateContributor

[aggregateAttributes]	*string*	Route map against which to evalute the aggregate route.
aggregateRole	*contributor*	Role in contributor-aggregate relation.

2.127.1.3.2.3 AsPathLengthRange

end	*integer*	The maximum length of the as-path.
start	*integer*	The minimum length of the as-path.

2.127.1.3.2.4 CommInstancesRange

end	*integer*	The maximum number of communities.
start	*integer*	The minimum number of communities.

2.127.1.3.2.5 DebugInfo

[community]	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for match community lists.
[communityAddOrReplace] **	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for add/replace community lists. **Since: 4.25.1
[communityDelete] **	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for delete community lists. **Since: 4.25.1
[communityFilter] **	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for filter community lists. **Since: 4.25.1
[extCommunity]	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for extended community lists.
[extCommunityFilter] **	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for filter extended community lists. **Since: 4.25.1
[largeCommunity]	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for large community lists.
[largeCommunityFilter] **	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for filter large community lists. **Since: 4.25.1
[matchResult]	*enum: unsupported* | *failed* | *matched*	Result of the matcher.

2.127.1.3.2.6 AttributeDebug

[ignored] **	*string*	String representation of element that may be ignored. **Since: 4.23.0

[undefined]	*string*	String representation of undefined element.

2.127.1.3.2.7 PolicyEvalSequenceResult

[continueSeqno]	*integer*	Continue to sequence number.
result	*enum: permit | fallThrough | deny*	Route map sequence result.

2.127.1.3.2.8 SetRules

[aigpMetric] **	*AigpMetric* see 2.127.1.3.2.9	AIGP Metric. **Since: 4.24.0
[asPathPrepend]	*string*	BGP AS path to prepend.
[asPathPrependLastAs]	*integer*	Number of times to prepend the last AS number.
[asPathReplace]	*string*	BGP AS path to replace.
[bpAsPathWeight] **	*integer*	BGP bestpath AS path weight parameter. **Since: 4.23.0
[communityAddOrReplace]	*CommunitySet* see 2.127.1.3.2.10	Community addor replace modifications.
[communityDelete]	*CommunitySet* see 2.127.1.3.2.10	Community deletemodifications.
[communityFilter] **	*CommunitySet* see 2.127.1.3.2.10	Community filtermodifications. **Since: 4.25.1
[debugInfo] **	*DebugInfo* see 2.127.1.3.2.5	Debugging information for set rules. **Since: 4.22.1
[distance]	*integer*	Protocol independent administrative distance.
[extCommunity]	*CommunitySet* see 2.127.1.3.2.10	Extended communitymodifications.
[extCommunityFilter] **	*CommunitySet* see 2.127.1.3.2.10	Extended communityfilter modifications. **Since: 4.25.1
[igpMetric]	*integer*	IGP metric.
[isisLevel]	*enum: level-1-2 | level-2 | level-1*	IS-IS route level.
[largeCommunity]	*CommunitySet* see 2.127.1.3.2.10	Large communitymodifications.
[largeCommunityFilter] **	*CommunitySet* see 2.127.1.3.2.10	Large communityfilter modifications. **Since: 4.25.1
[localPref]	*integer*	BGP local preference metric.
[nextHop]	*NextHop* see 2.127.1.3.2.11	Next hop information.
[origin]	*enum: bgpEGP | bgpIGP | bgpIncomplete*	BGP origin of path information.
[originAsValidity] **	*enum: valid | notFound | invalid*	AS Origin validation state. **Since: 4.23.2
[ospfBitDn] **	*OspfBit* see 2.127.1.3.2.12	DN bit to prevent routing loops. **Since: 4.25.1
[ospfMetricType]	*enum: type-2 | type-1*	OSPF external LSA metric type.
[ospfv3BitDn] **	*Ospfv3Bit* see 2.127.1.3.2.13	DN bit to prevent routing loops in ospfv3. **Since: **currentRelease**

[resolutionRibProfile-Config] **	*ResolutionRibProfileConfig* see 2.127.1.3.2.14	Resolution RIBs. **Since: 4.25.1
[routeMetric]	*RouteMetric* see 2.127.1.3.2.16	Route Metric.
[routeTag]	*integer*	Route tag.
[segmentIndex]	*integer*	MPLS Segment Routing Index.
[weight]	*integer*	BGP weight parameter.

2.127.1.3.2.9 AigpMetric

[metricType]	*enum: igp-next-hop-cost* | *metric* | *igp-metric*	AIGP metric from IGP metric or IGP next-hop-cost.
value	*integer*	AIGP metric value.

2.127.1.3.2.10 CommunitySet

[communityFilterTypes] **	*list of string*	Community filter type values. **Since: 4.25.1
[communityLists]	*list of string*	Community list names.
[communityValues]	*list of string*	Community list values.
operationType	*enum: filter* | *add* | *default* | *delete* | *none*	Community set operation type.

2.127.1.3.2.11 NextHop

[ipAddress]	*IP address*	Next hop IP address.
nextHopType	*enum: unchanged* | *ipv6Unchanged* | *peerAddress* | *ipAddress* | *ipv6PeerAddress* | *evpnUnchanged*	Next hop type.

2.127.1.3.2.12 OspfBit

ospfBitType	*dn*	Set an OSPF protocol bit.

2.127.1.3.2.13 Ospfv3Bit

ospfv3BitType	*dn*	Set an OSPFv3 protocol bit.

2.127.1.3.2.14 ResolutionRibProfileConfig

resolutionMethods	*list of ResolutionRib* see 2.127.1.3.2.15	List of resolution methods.
[systemDefault]	*boolean*	System default profile is set for next-hop.

2.127.1.3.2.15 ResolutionRib

[colored]	*boolean*	The tunnel RIB is colored.
name	*string*	Name of the resolution RIB.
ribType	*enum: tunnel | ip*	Type of the resolution RIB.

2.127.1.3.2.16 RouteMetric

[metricType]	*enum: add-igp-metric | subtractive | metric | multiplicative | igp-metric | additive | value-add-igp-nexthop-cost | igp-nexthop-cost | value-add-igp-metric | add-igp-nexthop-cost*	Additive or subtractive metric value.
value	*integer*	Route metric value.

2.127.1.4 RcfDebugEvaluation

entryPoint	*string*	The initial RCF expression that was invoked.
functionEvaluations	*table(string) = RcfDebugFunction* see 2.127.1.4.1	Evaluation information for each function invoked during the RCF evaluation keyed by function name.
result	*RcfDebugResult* see 2.127.1.4.1.5	The overall result of the RCF evaluation.

2.127.1.4.1 RcfDebugFunction

builtinFunction	*boolean*	Whether this function is builtin as part of theRCF language or user defined. A builtin function hasno definition, no code unit name, and no invocationsdetailed.
[codeUnitName]	*string*	Code unit containing the function definition.
invocations	*list of RcfDebugFunctionInvocation* see 2.127.1.4.1.1	A function can be invoked multiple times during RCF evaluation. The evaluation of the called function may differ each time. This is a list of all of the invocationsthat occurred.

2.127.1.4.1.1 RcfDebugFunctionInvocation

callStackDepth	*integer*	How deep in the call stack this invocation was evaluated.
fragments	*list of RcfDebugFragment* see 2.127.1.4.1.2	Fragments representing pieces of code relevant to the evaluation of the invoked function. Sorted first by their location (line and column number) and then by the order they occurred.

result	*RcfDebugResult* see 2.127.1.4.1.5	Result of the function invocations evaluation.

2.127.1.4.1.2 RcfDebugFragment

continuation	*boolean*	Whether this fragment is for an expression continuing from a previous line.
fragmentType	*enum: openExplicitBracket* | *closeBlock* | *statement* | *ifStatement* | *elseStatement* | *assignment* | *openBlock* | *functionCall* | *openImplicitBracket* | *closeStatementExpression* | *openStatementExpression* | *logicalOperator* | *closeImplicitBracket* | *definition* | *condition* | *closeExplicitBracket*	Type of code the fragment represents.
[invocation]	*RcfDebugFunction-InvocationReference* see 2.127.1.4.1.4	Reference to the RCF invocation that occurred in the code fragment.
location	*RcfDebugFragment-Location* see 2.127.1.4.1.3	Location the fragment occurred at in the RCF code.
[result]	*RcfDebugResult* see 2.127.1.4.1.5	Result of the code fragments evaluation.

2.127.1.4.1.3 RcfDebugFragmentLocation

column	*integer*	Column number of the location on a line.
[length]	*integer*	Length of the fragment in bytes.
line	*integer*	Line number of the location within a code unit.

2.127.1.4.1.4 RcfDebugFunctionInvocationReference

functionName	*string*	Name of the RCF function that was invoked.
invocationIndex	*integer*	A function can be invoked multiple times during RCF evaluation. The evaluation of the called function may differ each time. This index is used to indicate the particular invocation being referenced.

2.127.1.4.1.5 RcfDebugResult

[reason]	*string*	Explanation of the result.
value	*enum: true* | *false* | *unknown*	Trilean result value.

2.127.2 RcfCodeUnitText

functions	*table(string) = RcfFunctionText* see 2.127.2.1	Reference to the text of RCF functions where evaluation occurred, keyed by the RCF function name.

2.127.2.1 RcfFunctionText

lines	*table(numeric string) = string*	Lines of RCF text. Keyed by line number.

2.128 show bgp debug policy export evpn (PREFIX | PREFIX6) [...]

revision 2

see 2.127

2.129 show bgp debug policy import ((((vpn-route filter-rcf vpn-ipv4) | (ipv4 SAFI)) PREFIX) | (((vpn-route filter-rcf vpn-ipv6) | (ipv6 SAFI)) PREFIX6)) rd RD [...]

revision 2

see 2.127

2.130 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...]

revision 2

see 2.127

2.131 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) (vpn-ipv4 | vpn-ipv6) (PREFIX | PREFIX6) rd RD [...]

revision 2

see 2.127

2.132 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) evpn route-type ip-prefix (PREFIX | PREFIX6) rd RD [...]

revision 2

see 2.127

2.133 show bgp debug policy inbound neighbor (ADDR | ADDR6 | all) flow-spec (ipv4 | ipv6) rule FLOWSPEC_RULE [...]

revision 2

see 2.127

2.134 show bgp debug policy outbound neighbor (ADDR | ADDR6) (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...]

revision 2

see 2.127

2.135 show bgp debug policy outbound neighbor (ADDR | ADDR6) (vpn-ipv4 | vpn-ipv6) (PREFIX | PREFIX6) rd RD [...]

revision 2

see 2.127

2.136 show bgp debug policy outbound neighbor (ADDR | ADDR6) evpn route-type ip-prefix (PREFIX | PREFIX6) rd RD [...]

revision 2

see 2.127

2.137 show bgp debug policy outbound neighbor (ADDR | ADDR6) flow-spec (ipv4 | ipv6) rule FLOWSPEC_RULE [...]

revision 2

see 2.127

2.138 show bgp debug policy redistribute (ipv4 SAFI PREFIX) | (ipv6 SAFI PREFIX6) [...]

revision 2

see 2.127

2.139 show bgp evpn (route-type ((auto-discovery) | (mac-ip) | imet | (ethernet-segment) | (ip-prefix (ipv4 | ipv6 | PREFIX_V4 | PREFIX_V6)) | (smet | join-sync | leave-sync | spmsi))) internal [...]

revision 2

asn	*integer*	Autonomous System Number.
[currentSeqNum]	*integer*	Current route sequence number.
evpnRoutes	*table(string) = EvpnRouteEntry* see 2.139.1	Dictionary of BGP EVPN route entries indexed by the route key.
[lastProcessedSeqNum]	*integer*	Last route sequence number acknowledged.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF name.

2.139.1 EvpnRouteEntry

evpnRoutePaths	*list of EvpnRoutePath* see 2.139.1.2	List of BGP EVPN route ECMP paths.
routeKeyDetail	*EvpnRouteKeyDetail* see 2.139.1.1	NLRI details.
[totalPaths]	*integer*	Total number of paths for this route.

2.139.1.1 EvpnRouteKeyDetail

[domain] **	*enum: remote | local*	BGP route domain. **Since: 4.26.0
[esi]	*string*	For auto-discovery and ethernet-segment routes, the ethernet segment identifier.
[eti]	*integer*	For auto-discovery, mac-ip, imet, and ip-prefix routes, the ethernet tag identifier.
[eti2] **	*integer*	For dual-encap mac-ip and imet routes, the second ethernet tag identifier. **Since: 4.27.1
[group] **	*IP address*	Multicast group address. **Since: 4.23.0
[ipGenAddr]	*IP address*	For mac-ip routes, the IP address corresponding to the ethernet address. For imet and ethernet-segment routes, the originating router IP address.
[ipGenAddr2] **	*IP address*	For dual-encap imet routes, the second originating router IP address. **Since: 4.27.1
[ipGenPrefix]	*IP address prefix*	For ip-prefix routes, the IPv4 or IPv6 address prefix.
[mac]	*string*	For mac-ip routes, the ethernet address.

nlriType	*enum: leave-sync | imet | ethernet-segment | spmsi | smet | join-sync | auto-discovery | mac-ip | ip-prefix*	NLRI type.
[rd]	*string*	Route distinguisher.
[source] **	*IP address*	Multicast source address. **Since: 4.23.0
[syncnum] **	*integer*	Leave group synchronization number. **Since: 4.23.0

2.139.1.2 EvpnRoutePath

asPathEntry	*EvpnRouteASPathEntry* see 2.139.1.2.2	AS path information.
[importedEvpnPathRd]	*string*	Imported EVPN path RouteDistinguisher.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[nextHop]	*string*	Route next hop address.
reasonNotBestpath	*enum: ecmpFast | routerIdTB | noBest | ebgp | med | noLocalLabel | age | clusterlistLenTB | originatorId | macMobilitySeqNum | nexthopVrf | routeUnusable | aspathDetails | neighborMultipath | infMed | pathId | noReason | macMobilityTep-AddrdomainPathLength | macMobilityStickyMac | rrClient | routeNotReady | originAsValidity | igpCost | macMobilityRemoteDomain | nhgTunnelEcmpDisabled | aigp | originatorIdOrRouterId | origin | localpref | preference | clusterlistLen | aspathLength | macMobilityEsi | weight | ageTB | peerAddress | invalid | routerId | localPath | rtMembershipDefault-Originate | redist | originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*EvpnRouteDetailEntry* see 2.139.1.2.3	Route details.
routeType	*BgpVpnRouteTypeEntry* see 2.139.1.2.1	Route type.
[tag] **	*integer*	Tag for the route. **Since: 4.29.1

[timestamp]	*integer*	UTC seconds since epoch when the route was received. Only returned with 'show bgp evpn detail'.
[weight]	*integer*	Weight for the route.

2.139.1.2.1 BgpVpnRouteTypeEntry

active	*boolean*	Route is active.
atomicAggregator	*boolean*	Route is an atomic-aggregate.
backup	*boolean*	Route is backup.
ecmp	*boolean*	Route is an ECMP route.
ecmpContributor	*boolean*	Route contributes to ECMP.
ecmpHead	*boolean*	Route is the ECMP head.
[localAgg]	*boolean*	Route is locally aggregated.
[notInstalledReason]	*enum: labeledRoutePresent | routeInstallDenied | routeBestInactive*	Reason for route not being installed.
[origin]	*enum: Igp | Egp | Incomplete*	Route origin.
queued	*boolean*	Route is queued for advertisement.
stale	*boolean*	Route is stale.
suppressed	*boolean*	Route is suppressed from entering the RIB.
[ucmp]	*boolean*	Route is an UCMP route.
[unknownMetric]	*boolean*	Route is pending resolution.
valid	*boolean*	Route is valid.
[waitForConvergence]	*boolean*	Route is pending BGP convergence.

2.139.1.2.2 EvpnRouteASPathEntry

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
asPathType	*enum: Invalid | External | Confed-External | Internal | Local*	AS path type: Internal - originated by I-BGP External - originated by E-BGP Confed-External - originated by a E-BGP confederation Local - originated locally Invalid - AS path is invalid.

2.139.1.2.3 EvpnRouteDetailEntry

[aggregator]	*string*	Aggregator of the route.
[clusterList]	*string*	Cluster list for the route.
communities	*list of string*	List of communities for this route.
[domainPath] **	*list of string*	Domain path attribute. **Since: 4.23.0
[esi]	*string*	Ethernet segment identifier.
extCommunities	*list of string*	List of extended communities for this route.
extCommunityListRaw **	*list of integer*	List of raw extended communities for this route. **Since: 4.26.0

fromRRClient	*boolean*	Route received from route reflector client.
[gatewayIpAddr] **	*IP address*	Route gateway IP address. **Since: 4.29.0
[l3Label]	*EncapLabel* see 2.17.1.1.2.28	Local or remote L3 label.
[l3LabelMpls] **	*EncapLabel* see 2.17.1.1.2.28	Local or remote L3 MPLS label. **Since: 4.28.0
[l3LabelVxlan] **	*EncapLabel* see 2.17.1.1.2.28	Local or remote L3 VXLAN label. **Since: 4.28.0
[label]	*EncapLabel* see 2.17.1.1.2.28	Local or remote label.
[labelMpls] **	*EncapLabel* see 2.17.1.1.2.28	Local or remote MPLS label. **Since: 4.26.0
[labelSrv6] **	*EncapLabel* see 2.17.1.1.2.28	Local or remote SRv6 SID (label). **Since: 4.28.2
[labelStack]	*list of integer*	MPLS label stack information.
[labelVxlan] **	*EncapLabel* see 2.17.1.1.2.28	Local or remote VXLAN label. **Since: 4.26.0
largeCommunities **	*list of string*	List of large communities for route. **Since: 4.22.1
[multicastFlags] **	*MulticastFlags* see 2.139.1.2.3.1	Multicast flags for route-type smet, join-sync, and leave-sync. **Since: 4.26.2
origin	*enum: Igp | Egp | Incomplete*	Route origin.
[originator]	*IPv4 address*	Router ID of the originator of this route.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.
[pendingTimeToAdv]	*floating point number*	Timestamp of route advertisement.
[pmsiTunnel] **	*PmsiTunnel* see 2.17.1.1.2.31	P-Multicast Serivce Interface Tunnel Attribute. **Since: 4.23.2
[redistributionProtocol]	*enum: Connected | Ospf3 | Static | unknown | Is-Is | Rip | Ospf*	Protocol from which route got redistributed into BGP.
[rxPathId]	*integer*	Received path ID of this route.
[rxSafi]	*string*	Received SAFI of this route.
[seqNumber]	*integer*	Route sequence number.
[tunnelEncap] **	*table(string) = TunnelEncap* see 2.139.1.2.3.2	Tunnel encapsulation attribute. **Since: 4.25.0
tunnelRibEligible	*boolean*	Route eligible to be installed in Tunnel RIB.
[txPathId]	*integer*	Advertised path ID for this route.

2.139.1.2.3.1 MulticastFlags

[filterMode]	*enum: exclude | include*	Group membership filter mode.

2.139.1.2.3.2 TunnelEncap

asn	*integer*	Autonomous System Number.
endpoint	*IP address*	Tunnel egress endpoint.

2.140 show bgp evpn [...]

revision 2

see 2.139

2.141 show bgp evpn host-flap

duplicationBlacklistedMacs	*list of BgpEvpnHostFlapEntry* see 2.141.1	List of routes that contain MAC Addresses that are blacklisted due to duplication.

2.141.1 BgpEvpnHostFlapEntry

macAddr	*MAC address*	Blacklisted MAC Address due to duplication.
time	*floating point number*	Time when the MAC Address was added to the blacklist.
vlan	*integer*	VLAN ID of blacklisted MAC Address.

2.142 show bgp evpn instance [...]

revision 2

bgpEvpnInstances	*table(string) = BgpEvpnInstance* see 2.142.1	BGP EVPN instances indexed by instance name.

2.142.1 BgpEvpnInstance

[adLabel]	*integer*	MPLS label used in auto-discovery routes.
[afiSafiConfig] **	*table(string) = InstanceAfiSafiConfigs* see 2.142.1.3	Route targets for an address family type. **Since: 4.30.0
[ethernetSegments]	*table(string) = EthernetSegment* see 2.142.1.2	Local ethernet segments indexed by ESI.
[evpnVpwsDetail]	*BgpEvpnVpwsDetail* see 2.142.1.1	EVPN virtual private wire service (VPWS) detailed information.
[exportRemoteRts]	*list of integer*	Remote route targets to export.
[exportRts]	*list of integer*	Route targets to export.
[imetLabel]	*integer*	MPLS label used in imet routes.
[importRemoteRts]	*list of integer*	Remote route targets to import.
[importRts]	*list of integer*	Route targets to import.
[labelAllocMode]	*enum: perInstance | perPseudowire | perVlan*	Label allocation mode.
[localMplsIp]	*IP address*	Local MPLS IP.
[localVxlanIp]	*IP address*	Local VXLAN IP.
[macLabel]	*integer*	MPLS label used in mac-ip routes.
[mplsEnabled]	*boolean*	MPLS Enabled.
rd	*string*	EVPN instance route distinguisher.
[remoteEthernetSegments] **	*table(string) = RemoteEthernetSegment* see 2.142.1.4	Remote ethernet segments. **Since: 4.30.0
[remoteRd]	*string*	EVPN instance remote route distinguisher.
[serviceIntf]	*enum: vlanBased | vlanAwareBundle*	EVPN instance service interface.
[vxlanEnabled]	*boolean*	VXLAN Enabled.

2.142.1.1 BgpEvpnVpwsDetail

controlWord	*boolean*	Control word required when sending EVPN packets to this device.
[fxcNormalizedVlanTag-Signaling]	*enum: all | none*	Flexible cross connect normalized VLAN tag signaling.
[l2Mtu]	*integer*	L2 MTU value used for validation, if enabled.
pseudowires	*table(string) = BgpEvpnVpwsPseudowire* see 2.142.1.1.1	Mapping of pseudowire name to detailed information.

2.142.1.1.1 BgpEvpnVpwsPseudowire

[color] **	*integer*	Color. **Since: **currentRelease**
status	*enum: noLocalId | localIdConflict | noRemoteId | noLocalLabel | noRequest | pending | remoteIdConflict | up*	Pseudowire status.
[vlanTagNormalization]	*enum: double | single*	Flexible cross connect VLAN tag normalization.
[vpwsIdLocal]	*integer*	Advertised virtual private wire service (VPWS) instance identifier.
[vpwsIdRemote]	*integer*	Peer's virtual private wire service (VPWS) instance identifier.
[vpwsLabel]	*integer*	Advertised pseudowire label.

2.142.1.2 EthernetSegment

[dFElectionAlgorithm]	*enum: preference | hrw | modulus*	DF election algorithm.
[dFElectionState]	*enum: started | finished | pending*	DF election state.
dFPeer	*EsiPeer* see 2.142.1.2.1	Designated forwarder.
[esiLabel]	*integer*	MPLS ESI label used for split-horizon filtering.
forwardingPeers	*list of EsiPeer* see 2.142.1.2.1	All forwarding peers.
importRT	*string*	ES import route target.
intf	*interface name*	Interface name.
nonDFPeers	*list of EsiPeer* see 2.142.1.2.1	Non-designated forwarders.
[pseudowires]	*table(string) = EthernetSegmentPseudowire* see 2.142.1.2.2	Mapping of pseudowire name to detailed information.
redundancyMode	*enum: allActive | singleActive*	Redundancy mode.
[sharedEsiLabel]	*boolean*	MPLS ESI label shared with other PEs.
state	*enum: down | up*	Ethernet segment state.

2.142.1.2.1 EsiPeer

[esiLabel]	*integer*	MPLS ESI label advertised by peer.
ip	*IP address*	IP address of peer.

2.142.1.2.2 EthernetSegmentPseudowire

[backup]	*IP address*	IP address of the backup peer.

[primary]	*IP address*	IP address of the primary peer.

2.142.1.3 InstanceAfiSafiConfigs

[routeTargetExports]	*list of string*	Route targets to export.
[routeTargetImports]	*list of string*	Route targets to import.

2.142.1.4 RemoteEthernetSegment

activeTEPs	*list of IP address*	IP addresses of the remote peers that has the Ethernet segment.

2.143 show bgp evpn sanity [...]

Could also be an instance of:
EvpnSanityModelBrief 2.143.1.1
EvpnSanityModelDetail 2.143.1.1

categories	*table(string) = EvpnSanityCategory* see 2.143.1	EVPN sanity categories to verify, keyed by their name.

2.143.1 EvpnSanityCategory

description	*string*	Description of category.
items	*list of EvpnSanityItem* see 2.143.1.1	Items to verify in this category.

2.143.1.1 EvpnSanityItem

name	*string*	Name of item.
[reasonNotOk]	*string*	Reason why this check was not successful.
status	*enum: warn | ok | fail*	Status of verification of item.
categories	*table(string) = EvpnSanityCategory* see 2.143.1	EVPN sanity categories to verify, keyed by their name.
categories	*table(string) = EvpnSanityCategory* see 2.143.1	EVPN sanity categories to verify, keyed by their name.

2.144 show bgp evpn summary [...]

revision 3

see 2.46.1

2.145 show bgp flow-spec (ipv4 | ipv6 | vpn-ipv4 | vpn-ipv6) summary [...]

revision 3

see 2.46

2.146 show bgp flow-spec ipv4 [...]

vrfs	*table(string) = BgpFlowspecRules* see 2.146.1	Bgp Flowspec rule summary.

2.146.1 BgpFlowspecRules

asn	*integer*	Autonomous System Number.
routerId	*IPv4 address*	BGP Router Identity.
rules	*table(string) = BgpFlowspecRuleEntry* see 2.146.1.1	Dictionary of BGP Flowspec rule entries indexed by the route key.
vrf	*string*	VRF name.

2.146.1.1 BgpFlowspecRuleEntry

[matchDetails]	*list of string*	Descriptions of matching conditions.
paths	*list of BgpFlowspecRulePath* see 2.146.1.1.1	BGP flowspec rule paths.
ruleId	*integer*	Unique identifier of this rule.
totalPaths	*integer*	Total number of paths for this rule.

2.146.1.1.1 BgpFlowspecRulePath

actions	*list of BgpFlowspecAction* see 2.146.1.1.1.1	Actions that are performed for this rule.
asPathEntry	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[peerEntry] **	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route. **Since: 4.22.1

reasonNotBestpath	*enum: ecmpFast | routerIdTB | noBest | ebgp | med | noLocalLabel | age | clusterlistLenTB | originatorId | macMobilitySeqNum | nexthopVrf | routeUnusable | aspathDetails | neighborMultipath | infMed | pathId | noReason | macMobilityTep-AddrdomainPathLength | macMobilityStickyMac | rrClient | routeNotReady | originAsValidity | igpCost | macMobilityRemoteDomain | nhgTunnelEcmpDisabled | aigp | originatorIdOrRouterId | origin | localpref | preference | clusterlistLen | aspathLength | macMobilityEsi | weight | ageTB | peerAddress | invalid | routerId | localPath | rtMembershipDefault-Originate | redist | originatorIdTB*	Reason route was not selected as BGP best path.
[redirectVrf]	*BgpFlowspecRedirectVrf* see 2.146.1.1.1.2	VRF being used by a traffic redirect action.
[routeDetail]	*BgpRouteDetailEntry* see 2.17.1.1.2.2	Route details.
routeType	*BgpRouteTypeEntry* see 2.17.1.1.2.36	Route type.
[weight]	*integer*	Weight for the route.

2.146.1.1.1.1 BgpFlowspecAction

action	*enum: drop | markDscp | redirectVrf | police | mirrorIp | trafficAction | redirectIp*	Type of action that is performed.
[value]	*string*	Value associated with the action.

2.146.1.1.1.2 BgpFlowspecRedirectVrf

routeTarget	*string*	Route target that was used to select a VRF.
vrf	*string*	Name of the VRF traffic is being redirected to.

2.147 show bgp flow-spec ipv6 [...]

see 2.146

2.148 show bgp flow-spec vpn-ipv4 [...]

see 2.146

2.149 show bgp flow-spec vpn-ipv6 [...]

see 2.146

2.150 show bgp instance [...]

revision 2

vrfs	*table(string) = ShowBgpInstanceModel* see 2.150.1	per VRF bgp instance information models.

2.150.1 ShowBgpInstanceModel

adjRibInsAwaitingCleanup	*integer*	Number of Adj-RIB-Ins awaiting cleanup.
afiSafiConfig	*table(string) = ShowBgpInstanceAfiSafiConfigModel* see 2.150.1.1	Configuration keyed by AFI/SAFI name.
[afiSafiConvergence]	*table(string) = ShowBgpInstanceAfiSafiConvergenceModel* see 2.150.1.2	Convergence status keyed by AFI/SAFI name.
[afiSafiState]	*table(string) = ShowBgpInstanceAfiSafiStatusModel* see 2.150.1.3	Status keyed by AFI/SAFI name.
[bgpIpv4ListenPort]	*integer*	Port BGP is listening on IPv4.
bgpIpv4Listening	*boolean*	BGP is listening on IPv4.
[bgpIpv6ListenPort]	*integer*	Port BGP is listening on IPv6.
bgpIpv6Listening	*boolean*	BGP is listening on IPv6.
converged	*boolean*	BGP has converged.
[convergenceTimeout]	*integer*	Convergence timeout in seconds.
convergenceTimer	*boolean*	Convergence timer enabled.
[convergenceTimerExpires]	*floating point number*	Convergence expiration time since epoch.
eorTimeout	*integer*	End-of-RIB timeout in seconds.
fourByteAsn	*boolean*	four-byte ASN enabled.
gracefulRestart	*boolean*	Graceful restart mode enabled.
gracefulRestartHelper	*boolean*	Graceful restart helper mode.
gracefulRestartTimeout	*integer*	Graceful restart timeout in seconds.
localAs	*integer*	Local AS number.
[nexthopLuOriginateLfibUninstallDelay] **	*integer*	Next hop labeled unicast origination LFIB uninstall delay timer in seconds. **Since: 4.26.2
outstandingEors	*integer*	Number of outsdanding End-of-RIB's.
outstandingKeepalives	*integer*	Number of outstanding keepalives.
peerMacResolutionTimeout	*integer*	Peer MAC resolution in seconds.
peers	*ShowBgpInstancePeersModel* see 2.150.1.4	Peer statistics.
reflectedRoutesAttributesState	*enum: preserved | notPreserved | alwaysPreserved | unknown*	State of attributes of the reflected routes.
routerId	*IPv4 address*	Router ID.
slowPeerTimeout	*integer*	Slow peer timeout in seconds.
[timeToConverge]	*floating point number*	Time taken to BGP convergence in seconds.
[ucmpDeviation]	*floating point number*	UCMP deviation.

[ucmpLinkBwDelay]	*integer*	UCMP link bandwidth delay.
[ucmpMaximumPaths]	*integer*	UCMP maximum paths.
ucmpMode	*enum: ucmpMode1 | ucmpDisabled*	UCMP mode.

2.150.1.1 ShowBgpInstanceAfiSafiConfigModel

[additionalPaths-Installation]	*boolean*	Additional paths installation enabled.
[additionalPaths-InstallationEcmpPrimary]	*boolean*	Allow additional path with ECMP primary path.
[aigpSessionConfed]	*boolean*	AIGP session is enabled for confed peers.
[aigpSessionEbgp]	*boolean*	AIGP session is enabled for eBGP peers.
[aigpSessionIbgp]	*boolean*	AIGP session is enabled for iBGP peers.
[extendedNextHopCapability]	*boolean*	Extended next-hop capability.
[importedDatabases]	*table(string) = BgpLsImportDbEntry* see 2.150.1.1.1	Dictionary of BGP Link State imported LSDBs keyed by IGP instance name.
[ipLookupLocalLabel]	*string*	Local IP lookup MPLS VRF label.
[ipLookupLocalSid] **	*Srv6Sid* see 2.17.1.1.2.29	Local IP lookup SRv6 VRF SID. **Since: 4.28.2
[ipLuRibRouteMap]	*string*	IP LU RIB route map.
[ipLuTargetRibs]	*string*	IP LU Target RIBs.
[ipLuTunnelRibRouteMap]	*string*	IP LU tunnel RIB route map.
[lfibInstall] **	*boolean*	LFIB entry installation is enabled. **Since: **currentRelease**
[luOriginateEnabled-Nexthops] **	*list of string*	Labeled unicast originate enabled next hops. **Since: 4.26.0
[mplsResolutionRibs]	*NhResolutionRibProfile* see 2.150.1.1.2	MPLS resolution RIBs.
[rcfExports] **	*table(string) = string*	RCF function to apply on export, keyed by AFI/SAFI name. **Since: 4.27.2
[rcfImports] **	*table(string) = string*	RCF function to apply on import, keyed by AFI/SAFI name. **Since: 4.27.0
[redistributedRoutes]	*list of ShowBgpInstance-RedistProtoModel* see 2.150.1.1.4	Redistributed routes into BGP.
[resolutionRibs]	*NhResolutionRibProfile* see 2.150.1.1.2	Resolution RIBs.
[roles]	*list of string*	Link State roles and protocols. Can be any combination of 'producer', 'propagator', 'consumer' roles and 'IS-IS', 'DPS' protocols.
[routeDistinguisher]	*integer*	Route distinguisher.
[routeDistinguisherRemote-Domain] **	*integer*	Remote domain route distinguisher. **Since: 4.28.1
[routeMapExports]	*table(string) = string*	Route maps to apply on export, keyed by AFI/SAFI name. (overridden if RCF is configured).

[routeMapImports]	*table(string) = string*	Route maps to apply on import, keyed by AFI/SAFI name. (overridden if RCF is configured).
[routeTargetExportRemote-Domain] **	*ShowBgpInstanceAfiSafi-RouteLists* see 2.150.1.1.3	Remote domain route targets to export. **Since: 4.28.1
[routeTargetExports]	*ShowBgpInstanceAfiSafi-RouteLists* see 2.150.1.1.3	Route targets to export.
[routeTargetImportRemote-Domain] **	*ShowBgpInstanceAfiSafi-RouteLists* see 2.150.1.1.3	Remote domain route targets to import. **Since: 4.28.1
[routeTargetImports]	*ShowBgpInstanceAfiSafi-RouteLists* see 2.150.1.1.3	Route targets to import.
[sixPeResolutionRibs]	*NhResolutionRibProfile* see 2.150.1.1.2	6PE resolution RIBs.
[vxlanResolutionRibs]	*NhResolutionRibProfile* see 2.150.1.1.2	VXLAN resolution RIBs.
[waitForConvergence]	*boolean*	Convergence based update synchornization enable.

2.150.1.1.1 BgpLsImportDbEntry

source	*isis*	Source protocol.
status **	*enum: igpInstNotPresent | active | idConflict*	Status of source protocol's instance. **Since: 4.26.0
vrf	*string*	VRF of source protocol's instance.

2.150.1.1.2 NhResolutionRibProfile

[defaults]	*list of ResolutionRib* see 2.127.1.3.2.15	List of default resolution methods.
[error] **	*string*	Invalid resolution profile reason. **Since: 4.25.2
[resolutionMethods]	*list of ResolutionRib* see 2.127.1.3.2.15	List of resolution methods.
[routeMap]	*string*	Route map to determine resolution profile.

2.150.1.1.3 ShowBgpInstanceAfiSafiRouteLists

[dps]	*list of string*	List of routes for dynamic path selection.
[evpn]	*list of string*	List of routes for EVPN.
[linkState]	*list of string*	List of routes for BGP Link State.
[mplsVpnV4u]	*list of string*	List of routes for IPv4 MPLS VPN.
[mplsVpnV6u]	*list of string*	List of routes for IPv6 MPLS VPN.
[v4Flowspec]	*list of string*	List of routes for IPv4 Flowspec.
[v4SrTe]	*list of string*	List of routes for IPv4 SR-TE.
[v4lu]	*list of string*	List of routes for IPv4 Labeled Unicast.
[v4m]	*list of string*	List of routes for IPv4 Multicast.

[v4u]	*list of string*	List of routes for IPv4 Unicast.
[v6Flowspec]	*list of string*	List of routes for IPv6 Flowspec.
[v6SrTe]	*list of string*	List of routes for IPv6 SR-TE.
[v6lu]	*list of string*	List of routes for IPv6 Labeled Unicast.
[v6m]	*list of string*	List of routes for IPv6 Multicast.
[v6u]	*list of string*	List of routes for IPv6 Unicast.

2.150.1.1.4 ShowBgpInstanceRedistProtoModel

[includeLeaked]	*boolean*	Include leaked routes in redistribution.
proto	*string*	Source protocol name.
[routeMap]	*string*	Route map name.

2.150.1.2 ShowBgpInstanceAfiSafiConvergenceModel

converged	*boolean*	AFI/SAFI converged.
outstandingEors	*integer*	Number of outstanding End-of-RIB's.
outstandingKeepalives	*integer*	Number of outstanding Keepalives.
[timeToConverge]	*floating point number*	Time taken to converge.

2.150.1.3 ShowBgpInstanceAfiSafiStatusModel

[producerStats]	*ShowBgpInstanceProducer-StatsModel* see 2.150.1.3.1	Producer statistics.

2.150.1.3.1 ShowBgpInstanceProducerStatsModel

numIpv4Prefixes	*integer*	Number of LSDB IPv4 prefixes.
numIpv6Prefixes	*integer*	Number of LSDB IPv6 prefixes.
numLinks	*integer*	Number of LSDB links.
numNodes	*integer*	Number of LSDB nodes.

2.150.1.4 ShowBgpInstancePeersModel

dynamicPeers	*integer*	Number of dynamic peers.
establishedPeers	*integer*	Number of established peers.
staticPeers	*integer*	Number of static peers.
totalPeers	*integer*	Number of peers.

2.151 show bgp labeled-unicast forwarding [...]

entries	*table(numeric string) = BgpLuForwardingTunnelTableEntry* see 2.151.1	BGP LU Forwarding entries keyed by tunnel index.

2.151.1 BgpLuForwardingTunnelTableEntry

labels	*list of string*	Label stack.
vias	*list of Via* see 2.151.1.1	List of nexthops.

2.151.1.1 Via

Could also be an instance of:
IpVia 2.151.1.2
MplsTunnelVia 2.151.1.5
MplsVia 2.151.1.3
TunnelVia 2.151.1.6
TunnelFibVia 2.151.1.4

type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.151.1.2 IpVia

interface	*interface name*	Egress L3 interface of the next-hop.
nexthop	*IP address*	Nexthop IP address.
type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.151.1.3 MplsVia

[backupTunnelInfo] **	*TunnelInfo* see 2.151.1.3.1	Backup Tunnel Information. **Since: 4.23.0
interface	*interface name*	Egress L3 interface of the next-hop.
labels	*list of string*	Label stack.
nexthop	*IP address*	Nexthop IP address.
type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.151.1.3.1 TunnelInfo

[staticInterfaceTunnelInfo] **	*IpTunnelInfo* see 2.151.1.3.1.1	Tunnel encap information for a static interface tunnel. **Since: 4.21.0
[tunnelAddressFamily]	*enum: IPv6 | IPv4*	Address family for MPLS tunnels.
[tunnelAlgorithm] **	*string*	Flexible algorithm of the tunnel. **Since: 4.26.2
[tunnelColor] **	*integer*	Tunnel color. **Since: 4.29.2
[tunnelEndPoint]	*IP address prefix*	Route prefix for MPLS underlay nexthop resolution.

tunnelIndex	*integer*	Tunnel table index.
[tunnelName] **	*string*	Tunnel name. **Since: 4.26.2
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Nexthop | IS-IS SR | EosSdk sBFD | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | RSVP FRR | RSVP LER | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.
[tunnelVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of tunnel vias. **Since: 4.22.0

2.151.1.3.1.1 IpTunnelInfo

tunnelDestination	*IP address*	Location in the IP network where the IP tunnel terminates.
[tunnelDscp]	*integer*	Differentiated services code point.
tunnelEncap	*enum: Invalid | VOQ | MPLS | SRv6 | IPsec | GREoverIPsec | IP-in-IP | GRE | GUE*	Tunnel encapsulation type.
[tunnelHoplimit]	*integer*	Hop limit.
[tunnelKey]	*integer*	GRE IP tunnel key.
tunnelSource	*IP address*	Location in the IP network where the IP tunnel starts.
[tunnelTos]	*integer*	Type of service.
[tunnelTtl]	*integer*	Time to live.

2.151.1.3.1.2 TunnelViaInfo

[interface]	*interface name*	Egress L3 interface of next hop.
[interfaceDescription]	*string*	Interface description.
[labelStack]	*list of integer*	MPLS label stack (top-of-stack label first).
[nextHop] **	*string*	Next-Hop name. **Since: 4.22.0
[nexthopAddr]	*IP address*	Next hop IP address.
[nhgName] **	*string*	Next-Hop-Group name. **Since: 4.22.0
[resolvingTunnel] **	*TunnelId* see 2.151.1.3.1.4	Resolving tunnel information. **Since: 4.23.1
[srTePolicyId] **	*SrTePolicyId* see 2.151.1.3.1.3	SR-TE policy id. **Since: 4.26.0

2.151.1.3.1.3 SrTePolicyId

color	*integer*	Color of SR-TE policy tunnel.
endpoint	*IP address*	IP address of the SR-TE policy tunnel endpoint.

2.151.1.3.1.4 TunnelId

index	*integer*	Tunnel index per tunnel type.
type	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.151.1.4 TunnelFibVia

[encapId] **	*integer*	Encapsulation identifier. **Since: 4.21.3
[greEncap]	*GreTunnelInfo* see 2.151.1.4.1	Encapsulation parameters for GRE tunnel.
interface	*interface name*	Egress L3 interface of the next-hop.
[ipInIpEncap] **	*IpInIpTunnelInfo* see 2.151.1.4.3	Encapsulation parameters for IP-in-IP tunnel. **Since: 4.29.1
[ipsecEncap]	*IpsecTunnelInfo* see 2.151.1.4.4	Encapsulation parameters for IPsec tunnel.
[ipsecGreEncap] **	*IpsecGreTunnelInfo* see 2.151.1.4.2	Encapsulation parameters for GRE over IPsec tunnel. **Since: 4.21.3
[mplsEncap]	*MplsTunnelInfo* see 2.151.1.4.5	Encapsulation parameters for MPLS tunnel.
nexthop	*IP address*	Nexthop IP address.
[nhgId] **	*integer*	Nexthop Group Id for NHG tunnel. **Since: 4.26.2
[nhgName] **	*string*	Nexthop Group name for NHG tunnel. **Since: 4.22.0

[resolvingTunnel] **	*TunnelId* see 2.151.1.3.1.4	Resolving tunnel Information. **Since: 4.23.1
[srTePolicyId] **	*SrTePolicyId* see 2.151.1.3.1.3	SR-TE policy information. **Since: 4.26.0
[srv6Encap] **	*Srv6TransportTunnelInfo* see 2.151.1.4.6	Encapsulation parameters for SRv6 transport tunnel. **Since: 4.29.0
type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.151.1.4.1 GreTunnelInfo

Could also be an instance of:
IpsecGreTunnelInfo 2.151.1.4.2

checksum	*boolean*	Checksum in use.
[dscp]	*integer*	Differentiated services code point.
dstAddr	*IP address*	Location in the IP network where the tunnel terminates.
[hoplimit]	*integer*	Hop limit.
[key]	*integer*	GRE key.
sequence	*boolean*	Sequence numbers in use.
srcAddr	*IP address*	Location in the IP network where the tunnel starts.
[tos]	*integer*	Type of service.
[ttl]	*integer*	Time to live.

2.151.1.4.2 IpsecGreTunnelInfo

checksum	*boolean*	Checksum in use.
[dscp]	*integer*	Differentiated services code point.
dstAddr	*IP address*	Location in the IP network where the tunnel terminates.
[hoplimit]	*integer*	Hop limit.
[key]	*integer*	GRE key.
sequence	*boolean*	Sequence numbers in use.
srcAddr	*IP address*	Location in the IP network where the tunnel starts.
[tos]	*integer*	Type of service.
[ttl]	*integer*	Time to live.

2.151.1.4.3 IpInIpTunnelInfo

[dscp]	*integer*	Differentiated services code point.
dstAddr	*IP address*	Location in the IP network where the tunnel terminates.
[hoplimit]	*integer*	Hop limit.
srcAddr	*IP address*	Location in the IP network where the tunnel starts.
[tos]	*integer*	Type of service.

[ttl]	*integer*	Time to live.

2.151.1.4.4 IpsecTunnelInfo

[dscp]	*integer*	Differentiated services code point.
dstAddr	*IP address*	Location in the IP network where the tunnel terminates.
[hoplimit]	*integer*	Hop limit.
srcAddr	*IP address*	Location in the IP network where the tunnel starts.
[tos]	*integer*	Type of service.
[ttl]	*integer*	Time to live.

2.151.1.4.5 MplsTunnelInfo

labelStack	*list of string*	MPLS label stack.

2.151.1.4.6 Srv6TransportTunnelInfo

srcAddr	*IP address*	Location in the IP network where the tunnel starts.

2.151.1.5 MplsTunnelVia

labels	*list of string*	Label stack.
tunnelId	*TunnelId* see 2.151.1.3.1.4	Tunnel Identifier.
type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.151.1.6 TunnelVia

tunnelId	*TunnelId* see 2.151.1.3.1.4	Tunnel Identifier.
type	*enum: tunnel | ip*	Type of via, indicating the expected attributes.

2.152 show bgp labeled-unicast forwarding raw [...]

see 2.151

2.153 show bgp labeled-unicast tunnel [...]

revision 2

entries	*table(numeric string) = BgpLuTunnelTableEntry* see 2.153.1	BGP LU tunnel table entries keyed by tunnel index.

2.153.1 BgpLuTunnelTableEntry

bgpMetric	*integer*	BGP Metric for tunnel entry.
bgpMetric2	*integer*	BGP Metric 2 for tunnel entry.
[bgpMetricType] **	*enum: MED | metric | AIGP*	BGP Metric Type for tunnel entry. **Since: 4.26.0
bgpPref	*integer*	BGP Preference for tunnel entry.
bgpPref2	*integer*	BGP Preference 2 for tunnel entry.
contribution	*string*	Contribution of tunnel entry to Tunnel RIB.
endpoint	*IP address prefix*	Endpoint of the tunnel.
labels	*list of string*	Label stack.
vias	*list of Via* see 2.151.1.1	List of nexthops.

2.154 show bgp labeled-unicast tunnel raw [...]

revision 2

see 2.153

2.155 show bgp link-state [...]

revision 2

vrfs	*table(string) = BgpLsRoutes* see 2.155.1	BGP Link State routes.

2.155.1 BgpLsRoutes

asn	*string*	Autonomous System Number.
bgpLsRoutes	*table(string) = BgpLsRouteEntry* see 2.155.1.1	Dictionary of BGP Link State route entries keyed by a string representation of the NLRI.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF name.

2.155.1.1 BgpLsRouteEntry

identifier	*integer*	IGP Instance Identifier.
[linkDescriptor]	*BgpLsLinkDescriptor* see 2.155.1.1.1	Identity of link for link NLRIs.
localNodeDescriptor	*BgpLsNodeDescriptor* see 2.155.1.1.2	Identity of local node.
nlriType	*enum: node | link | ipv6Prefix | ipv4Prefix*	NLRI type.
paths	*list of BgpLsRoutePath* see 2.155.1.1.4	BGP-LS paths.
[prefixDescriptor]	*BgpLsPrefixDescriptor* see 2.155.1.1.3	Identity of prefix for IPv4 and IPv6 prefix NLRIs.
protocolId	*enum: isisLevel2 | isisLevel1 | bgpDps*	IGP source protocol.
[remoteNodeDescriptor]	*BgpLsNodeDescriptor* see 2.155.1.1.2	Identity of remote node for link NLRIs.
totalPaths	*integer*	Total number of paths for this BGP-LS route.

2.155.1.1.1 BgpLsLinkDescriptor

localIpv4Addrs	*list of IPv4 address*	List of local IPv4 interface addresses.
localIpv6Addrs	*list of IPv6 address*	List of local IPv6 interface addresses.
[localWanGroupId] **	*integer*	Local WAN group Identifier. **Since: 4.28.1
[localWanId] **	*integer*	Local WAN Identifier. **Since: 4.28.1
remoteIpv4Addrs	*list of IPv4 address*	List of remote IPv4 interface addresses.
remoteIpv6Addrs	*list of IPv6 address*	List of remote IPv6 interface addresses.
[remoteWanId] **	*integer*	Remote WAN Identifier. **Since: 4.28.1
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Link Descriptor Sub TLVs. **Since: 4.30.1

2.155.1.1.2 BgpLsNodeDescriptor

[asn]	*string*	Autonomous System Number of BGP-LS producer.
[bgpRouterId] **	*IPv4 address*	BGP router Identifier. **Since: 4.28.1
igpRouterIdType	*enum: isisSystemId | invalid | isisPseudoNodeId*	Type of IGP router identifier.
[isisPnodeId]	*string*	IS-IS pseudonode Identifier.
[isisSystemId]	*string*	IS-IS system Identifier.
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Node Descriptor Sub TLVs. **Since: 4.30.1

2.155.1.1.3 BgpLsPrefixDescriptor

prefix	*IP address prefix*	Prefix originated in IGP.
unknownTlvs **	*list of BgpLsUnknownTlv* see 2.17.1.1.2.6	List of unknown Prefix Descriptor Sub TLVs. **Since: 4.30.1

2.155.1.1.4 BgpLsRoutePath

asPathEntry	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
[localPreference] **	*integer*	I-BGP Local preference indicator. **Since: 4.30.0
[med]	*integer*	Multi Exit Discriminator.
[nextHop]	*string*	Next hop address.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information.

reasonNotBestpath	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*BgpRouteDetailEntry* see 2.17.1.1.2.2	Route details.
routeType	*BgpRouteTypeEntry* see 2.17.1.1.2.36	Route type.
[timestamp]	*integer*	UTC seconds since epoch when the path was received.
[weight]	*integer*	Weight for the path.

2.156 show bgp link-state summary

revision 3

see 2.46.1

2.157 show bgp monitoring [...]

bgpAdjRibInExportStatus	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
bmpGlobalConfig **	*BmpGlobalConfig* see 2.157.2	Bmp export policy and timestamp configuration. **Since: 4.21.3
stations	*table(string) = BmpStation* see 2.157.3	Collection of station information indexed by the station's name.

2.157.1 BgpAdjRibInExportStatus

monitoringEnabled	*boolean*	BGP Monitoring Enabled.

2.157.2 BmpGlobalConfig

afiSafiExport **	*table(string) = boolean*	Address family export enabled flag, keyed by address family name string. **Since: 4.23.0
exportIpv6LuTunnel **	*boolean*	Export IPv6 LU tunnel paths. **Since: 4.23.0
exportSixPe **	*boolean*	Export 6PE paths. **Since: 4.23.0
postPolicyExport	*boolean*	Export received post-policy routes.
prePolicyExport	*boolean*	Export received pre-policy routes.
timestampMode	*enum: none | sendTime*	BMP Timestamp setting: none - No timestamp set in BGP monitoring protocol message sendTime - Time when BGP monitoring protocol message is sent.

2.157.3 BmpStation

[address]	*StationHostnameOrAddress* see 2.157.3.3	The hostname or address of the station.
connected	*boolean*	The state of the connection to the station.
connectionMode	*enum: active | passive | unknown*	The mode used to connect to the station: unknown - No mode is configured, passive - Router listens for station to connect, active - Router initiates the connection to the station.
[description]	*string*	Station description.
[exportPolicyStationConfig] **	*BmpStationPolicies* see 2.157.3.1	BMP station export policy configuration. **Since: 4.27.1
flapCount	*integer*	Number station disconnects from the router.
[remotePort]	*integer*	The port of the station used for active connection mode.
state	*enum: connecting | idle | up*	The state of the station: idle - Station is stopped due to invalid config, connecting - Station is awaiting for the connection or attempting to connect, up - Station connection is established.

[tcpSocketStatistics] **	*BmpTcpStatistics* see 2.157.3.2	Peer TCP information. **Since: 4.20.5
[upTime]	*floating point number*	The timestamp in UTC when station transitioned to up.
vrfName	*string*	The VRF the station will be connected in.

2.157.3.1 BmpStationPolicies

postPolicyExport	*boolean*	This station exports received post-policy routes.
prePolicyExport	*boolean*	This station exports received pre-policy routes.

2.157.3.2 BmpTcpStatistics

congestionWindow	*integer*	TCP send congestion window (bytes).
delayedAckTimeout	*integer*	TCP socket delayed ack timeout (micro secions).
maxSegmentSize	*integer*	TCP outgoing maximum segment size (bytes).
options	*TcpOptions* see 2.157.3.2.1	TCP socket options.
outputQueueLength	*integer*	TCP output queue length.
outputQueueMaxLength	*integer*	TCP output queue max length.
recvRtt	*integer*	TCP receive round-trip time (micro seconds).
recvWindow	*integer*	TCP advertised receive window (bytes).
recvWindowScale	*integer*	TCP socket receive window scale factor.
retransmitTimeout	*integer*	TCP socket retransmission timeout (micro seconds).
sendRtt	*integer*	TCP round-trip time (micro seconds).
sendRttVariance	*integer*	TCP round-trip time variance (micro seconds).
sendWindowScale	*integer*	TCP socket send window scale factor.
slowStartThreshold	*integer*	TCP send slow start size threshold (bytes).
state	*enum: CLOSING | NOTUSED | LISTEN | SYN-RECEIVED | CLOSE-WAIT | FIN-WAIT-2 | CLOSED | LAST-ACK | ESTABLISHED | TIME-WAIT | FIN-WAIT-1 | SYN-SENT*	TCP Socket current state.
totalRetrans	*integer*	Total number of TCP retransmissions.

2.157.3.2.1 TcpOptions

ecn	*boolean*	Explicit Congestion Notification enabled.
selectiveAcks	*boolean*	Selective Acknowledgments enabled.
timestamps	*boolean*	Timestamps enabled.

windowScale	*boolean*	Window Scale enabled.

2.157.3.3 StationHostnameOrAddress

[hostName]	*string*	The hostname of the station.
[ip]	*IP address*	The IP address of the station.

2.158 show bgp monitoring active [...]

bgpAdjRibInExportStatus	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
stations	*table(string) = BmpActive* see 2.158.1	The collection of stations that are using active mode.

2.158.1 BmpActive

address	*StationHostnameOrAddress* see 2.157.3.3	The hostname or address of the station.
connected	*boolean*	The state of the connection to the station.
connectionAttempts	*integer*	The number of times the router has attempted to connect to the station.
connectionErrors	*integer*	The number of times the router has failed to connect to the station.
connectionSuccesses	*integer*	The number of times the router has successfully connected to the station.
[reason]	*string*	The reason why the connection is False.
remotePort	*integer*	The port of the station used for active connection mode.
retryInterval	*integer*	The interval in seconds which the connection will be retried if an attempt fails.
[retryTime]	*floating point number*	The timestamp in UTC of when the router will initiate a connection to the station.
vrfName	*string*	The VRF the station will be connected in.

2.159 show bgp monitoring active summary

bgpAdjRibInExportStatus	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
stations	*table(string) = BmpActiveSummary* see 2.159.1	The collection of stations that are using active mode.

2.159.1 BmpActiveSummary

connected	*boolean*	The state of the connection to the station.
[retryTime]	*floating point number*	The timestamp in UTC of when the router will initiate a connection to the station.

2.160 show bgp monitoring internal scheduler [...]

see 2.79

2.161 show bgp monitoring passive [...]

revision 2

bgpAdjRibInExportStatus	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
listenersV4	*table(string) = BmpPassive* see 2.161.1	Collection of IPv4 BMP listeners indexed by VRF name.
listenersV6	*table(string) = BmpPassive* see 2.161.1	Collection of IPv6 BMP listeners indexed by VRF name.

2.161.1 BmpPassive

connectionsAccepted	*integer*	Number of connections accepted.
connectionsRejected	*integer*	Number of connections rejected.
listening	*boolean*	The state of the listener for when the listener is ready to accept connections.
port	*integer*	Port number that is listening for connections.
[reason]	*string*	The reason why listening is False.
stations	*table(string) = StationHostnameOrAddress* see 2.157.3.3	The collection of stations and their hostname this listener will accept when a new connection is made indexed by the station's name.

2.162 show bgp monitoring passive summary

revision 2

bgpAdjRibInExportStatus	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
listenersV4	*table(string) = BmpPassiveSummary* see 2.162.1	Collection of IPv4 BMP listeners indexed by VRF name.
listenersV6	*table(string) = BmpPassiveSummary* see 2.162.1	Collection of IPv6 BMP listeners indexed by VRF name.

2.162.1 BmpPassiveSummary

listening	*boolean*	Listener is ready to accept connections.
port	*integer*	Port number that is listening for connections.
stationCount	*integer*	Number of stations configured for this listener.

2.163 show bgp monitoring summary

`bgpAdjRibInExportStatus`	*BgpAdjRibInExportStatus* see 2.157.1	Status of Rib Export for BMP.
`bmpGlobalConfig` **	*BmpGlobalConfig* see 2.157.2	Bmp export policy and timestamp configuration. **Since: 4.21.3
`stations`	*table(string) = BmpStationSummary* see 2.163.1	Collection of station information indexed by the station's name.

2.163.1 BmpStationSummary

`state`	*enum: connecting | idle | up*	The state of the station: idle - Station is stopped due to invalid config, connecting - Station is awaiting for the connection or attempting to connect, up - Station connection is established.
`[upTime]`	*floating point number*	The timestamp in UTC when station transitioned to up.

2.164 show bgp mvpn-ipv4 (vrf VRF) route-type intra-as-ipmsi [...]

vrfs	*table(string) = MvpnRoutes* see 2.164.1	MVPN route summary.

2.164.1 MvpnRoutes

asn	*integer*	Autonomous System Number.
mvpnRoutes	*table(string) = MvpnRouteEntry* see 2.164.1.1	Dictionary of MVPN route entries indexed by the route key.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF name.

2.164.1.1 MvpnRouteEntry

mvpnRoutePaths	*list of MvpnRoutePath* see 2.164.1.1.2	List of MVPN route ECMP paths.
routeKeyDetail	*MvpnRouteKeyDetail* see 2.164.1.1.1	NLRI details.
[totalPaths]	*integer*	Total number of paths for this route.

2.164.1.1.1 MvpnRouteKeyDetail

[group]	*IP address*	Multicast group address.
nlriType	*enum: intraAsIpmsi | selectivePmsi | sourceTree | sourceActive | sharedTree*	NLRI type.
[originator]	*IP address*	The originating routers IP address in intra-as-ipmsi routes.
[rd]	*string*	Route distinguisher. For locally sourced source-tree and shared-tree routes, set to the RD of the UMH.
[source]	*IP address*	Multicast source or RP address.
[sourceAs]	*string*	The originating routers Autonomous System in shared-tree and source-tree routes.

2.164.1.1.2 MvpnRoutePath

asPathEntry	*MvpnRouteAsPathEntry* see 2.164.1.1.2.1	AS path information.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[nextHop]	*string*	Route next hop address.

reasonNotBestpath	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*MvpnRouteDetailEntry* see 2.164.1.1.2.2	Route details.
routeType	*BgpVpnRouteTypeEntry* see 2.139.1.2.1	Route type.
[tag] **	*integer*	Tag for the route. **Since: 4.29.1
[timestamp]	*integer*	UTC seconds since epoch when the route was received.Only returned with detail output.
[weight]	*integer*	Weight for the route.

2.164.1.1.2.1 MvpnRouteAsPathEntry

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
asPathType	*enum: Invalid* | *External* | *Confed-External* | *Internal* | *Local*	AS path type: Internal: originated by I-BGP, External: originated by E-BGP, Confed-External: originated by a E-BGP confederation, Local: originated locally, Invalid: AS path is invalid.

2.164.1.1.2.2 MvpnRouteDetailEntry

[clusterList]	*string*	Cluster list for the route.
communities	*list of string*	List of communities for this route.

extCommunities	*list of string*	List of extended communities for this route.
extCommunityListRaw	*list of integer*	List of raw extended communities for this route.
fromRRClient	*boolean*	Route received from route reflector client.
largeCommunities	*list of string*	List of large communities for route.
origin	*enum: Igp | Egp | Incomplete*	Route origin.
[originator]	*IPv4 address*	Router ID of the originator of this route.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.
[pmsiTunnel]	*PmsiTunnel* see 2.17.1.1.2.31	P-Multicast Serivce Interface Tunnel Attribute.
[rxSafi]	*string*	Received SAFI of this route.
tunnelRibEligible	*boolean*	Route eligible to be installed in Tunnel RIB.

2.165 show bgp mvpn-ipv4 (vrf VRF) route-type shared-tree [...]

see 2.164

2.166 show bgp mvpn-ipv4 (vrf VRF) route-type source-active [...]

see 2.164

2.167 show bgp mvpn-ipv4 (vrf VRF) route-type source-tree [...]

see 2.164

2.168 show bgp mvpn-ipv4 route-type intra-as-ipmsi [...]

see 2.164

2.169 show bgp mvpn-ipv4 route-type shared-tree [...]

see 2.164

2.170 show bgp mvpn-ipv4 route-type source-active [...]

see 2.164

2.171 show bgp mvpn-ipv4 route-type source-tree [...]

see 2.164

2.172 show bgp neighbors (ADDR | ADDR6) rt-membership filter

`vrfs`	*table(string) = ShowRtMembership-NeighborFilterVrfModel* see 2.172.1	RT membership neighbor filter information, keyed by VRF name.

2.172.1 ShowRtMembershipNeighborFilterVrfModel

`peers`	*table(IPv4 or IPv6 address) = ShowRtMembership-NeighborFilterPeerModel* see 2.172.1.1	RT membership VRF neighbor filter information, keyed by peer address.

2.172.1.1 ShowRtMembershipNeighborFilterPeerModel

`routeTargets`	*list of route target*	RT membership VRF and neighbor route target filter.

2.173 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) ((routes | advertised-routes | (received-routes))) ((IP | IP_PREFIX)) [...]

revision 3

see 2.17

2.174 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) ((routes | advertised-routes | (received-routes))) [...]

revision 3

see 2.17

2.175 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) evpn (advertised-routes | routes | received-routes) [...]

revision 2

see 2.139

2.176 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) flow-spec ipv4 ROUTE_TYPE [...]

see 2.146

2.177 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) flow-spec ipv6 ROUTE_TYPE [...]

see 2.146

2.178 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) link-state ((routes | advertised-routes | (received-routes))) [...]

revision 2

see 2.155

2.179 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) path-selection ROUTE_TYPE [...]

vrfs	*table(string) = BgpDpsVrfRoutes* see 2.179.1	BGP path selection routes keyed by VRF.

2.179.1 BgpDpsVrfRoutes

asn	*string*	Autonomous system number.
nlris	*table(string) = BgpDpsVrfNlriRoutes* see 2.179.1.1	BGP path selection routes keyed by NLRI.
routerId	*IPv4 address*	BGP router identity.

2.179.1.1 BgpDpsVrfNlriRoutes

vteps	*table(IPv4 or IPv6 address prefix) = BgpDpsRouteEntry* see 2.179.1.1.1	BGP path selection routes keyed by VTEP address.

2.179.1.1.1 BgpDpsRouteEntry

dpsRoutePaths	*list of BgpDpsRoutePath* see 2.179.1.1.1.1	List of BGP path selection route ECMP paths.
[totalPaths]	*integer*	Total number of paths for this route.

2.179.1.1.1.1 BgpDpsRoutePath

[asPathEntry]	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
[avtNodeInfo]	*BgpDpsAvtNodeInfo* see 2.179.1.1.1.2	Adaptive virtual topology information.
[endpointAddr]	*IP address*	DPS endpoint address.
[ipSecurityInfo]	*IpSecurityInfo* see 2.179.1.1.1.4	IP security tunnel encapsulation information.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator.
[nextHop]	*string*	Next hop address.
[pathGroupName]	*string*	DPS path group name.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information.
routeState	*BgpDpsRouteState* see 2.179.1.1.1.3	Route state.
[timestamp]	*floating point number*	UTC seconds since epoch when the path was received.
[udpPort]	*integer*	DPS endpoint UDP port.
[wanId]	*integer*	DPS endpoint WAN ID.
[weight]	*integer*	Weight for the path.

2.179.1.1.1.2 BgpDpsAvtNodeInfo

[regionId]	*integer*	Adaptive virtual topology region ID.
[role]	*enum: zone transit* | *edge* | *unknown* | *pathfinder* | *region transit*	Adaptive virtual topology role.
[siteId]	*integer*	Adaptive virtual topology site ID.
[zoneId]	*integer*	Adaptive virtual topology zone ID.

2.179.1.1.1.3 BgpDpsRouteState

queued	*boolean*	Route is queued for advertisement.
stale	*boolean*	Route is stale.
valid	*boolean*	Route is valid.

2.179.1.1.1.4 IpSecurityInfo

algorithms	*list of IpSecurityAlgorithmInfo* see 2.179.1.1.1.5	Algorithm information.
[initialContact]	*boolean*	A flag to indicate whether this is an initial contact.
keyExchanges	*list of IpSecurityKeyExchangeInfo* see 2.179.1.1.1.6	Key exchange information.
[nonceLength]	*integer*	Length of nonce data in bytes.
[rekeyCounter]	*integer*	Rekey counter.

2.179.1.1.1.5 IpSecurityAlgorithmInfo

[authenticationAlgo]	*enum: sha256* | *sha512* | *sha384* | *sha1* | *nullhash* | *md5*	Authentication algorithm.
[encryptionAlgo]	*enum: nullesp* | *des* | *aes256* | *aes256gcm128* | *aes128gcm64* | *aes128gcm128* | *aes128*	Encryption algorithm.

2.179.1.1.1.6 IpSecurityKeyExchangeInfo

[dhGroup]	*integer*	Diffie-Hellman group number.
[dhKeyLength]	*integer*	Length of Diffie-Hellman public value in bytes.

2.180 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) rt-membership ROUTE_TYPE

vrfs	*table(string) = RtMembershipRoutes* see 2.180.1	RT MEMBERSHIP route summary.

2.180.1 RtMembershipRoutes

asn	*integer*	Autonomous System Number.
routerId	*IPv4 address*	BGP Router Identity.
rtMembershipRoutes	*table(string) = RtMembershipRouteEntry* see 2.180.1.1	Dictionary of BGP RT MEMBERSHIP route entries indexed by the route key.
vrf	*string*	VRF name.

2.180.1.1 RtMembershipRouteEntry

routeKeyDetail	*RtMembershipRouteKeyDetail* see 2.180.1.1.1	NLRI details.
rtMembershipRoutePaths	*list of RtMembershipRoutePath* see 2.180.1.1.2	List of BGP RT MEMBERSHIP route ECMP paths.
[totalPaths]	*integer*	Total number of paths for this route.

2.180.1.1.1 RtMembershipRouteKeyDetail

nlriLength	*integer*	The number of bits (0-96) in the overall Route Target membership NLRI.
originAs	*string*	The ASN of the Autonomous System that originated the Route Target prefix.
routeTargetPrefix	*RtMembershipRouteTargetPrefix* see 2.180.1.1.1.1	The Route Target prefix.

2.180.1.1.1.1 RtMembershipRouteTargetPrefix

maskLength	*integer*	The number of bits (0-64) in the Route Target prefix.
routeTarget	*integer*	The Route Target prefix value.

2.180.1.1.2 RtMembershipRoutePath

asPathEntry	*RtMembershipRouteASPathEntry* see 2.180.1.1.2.1	AS path information.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[nextHop]	*string*	Route next hop address.

[routeDetail]	*RtMembershipRouteDetail-Entry* see 2.180.1.1.2.2	Route details.
routeType	*RtMembershipRouteType-Entry* see 2.180.1.1.2.3	Route type.
[tag] **	*integer*	Tag for the route. **Since: 4.29.1
[timestamp]	*integer*	UTC seconds since epoch when the route was received.Only returned with detail output.
[weight]	*integer*	Weight for the route.

2.180.1.1.2.1 RtMembershipRouteASPathEntry

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
asPathType	*enum: Invalid* | *External* | *Confed-External* | *Internal* | *Local*	AS path type: Internal - originated by I-BGP External - originated by E-BGP Confed-External - originated by a E-BGP confederation Local - originated locally Invalid - AS path is invalid.

2.180.1.1.2.2 RtMembershipRouteDetailEntry

communities	*list of string*	List of communities for this route.
extCommunities	*list of string*	List of extended communities for this route.
extCommunityListRaw **	*list of integer*	List of raw extended communities for this route. **Since: 4.26.0
fromRRClient	*boolean*	Route received from route reflector client.
largeCommunities	*list of string*	List of large communities for this route.
origin	*enum: Igp* | *Egp* | *Incomplete*	Route origin.
[originator]	*IPv4 address*	Router ID of the originator of this route.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.

[reasonNotBestpath]	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[rxSafi]	*string*	Received SAFI of this route.

2.180.1.1.2.3 RtMembershipRouteTypeEntry

active	*boolean*	Route is active.
atomicAggregator	*boolean*	Route is an atomic-aggregate.
backup	*boolean*	Route is backup.
ecmp	*boolean*	Route is an ECMP route.
ecmpContributor	*boolean*	Route contributes to ECMP.
ecmpHead	*boolean*	Route is the ECMP head.
[localAgg]	*boolean*	Route is locally aggregated.
[origin]	*enum: Igp* | *Egp* | *Incomplete*	Route origin.
queued	*boolean*	Route is queued for advertisement.
stale	*boolean*	Route is stale.
suppressed	*boolean*	Route is suppressed from entering the Rib.
[ucmp]	*boolean*	Route is an UCMP route.
valid	*boolean*	Route is valid.
[waitForConvergence]	*boolean*	Route is pending BGP convergence.

2.181 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) sr-te POLICY_FILTER [...]

revision 3

see 2.17

2.182 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv4 ((routes | advertised-routes | (received-routes))) PREFIX (rd RD_VAL) internal [...]

revision 2

vrfs	*table(string) = MplsVpnRoutes* see 2.182.1	MPLS VPN route summary.

2.182.1 MplsVpnRoutes

asn	*integer*	Autonomous System Number.
[currentSeqNum]	*integer*	Current route sequence number.
[lastProcessedSeqNum]	*integer*	Last route sequence number acknowledged.
mplsVpnRoutes	*table(string) = MplsVpnRouteEntry* see 2.182.1.1	Dictionary of BGP MPLS VPN route entries indexed by the route key.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF name.

2.182.1.1 MplsVpnRouteEntry

mplsVpnRoutePaths	*list of MplsVpnRoutePath* see 2.182.1.1.2	List of BGP MPLS VPN route ECMP paths.
routeKeyDetail	*MplsVpnRouteKeyDetail* see 2.182.1.1.1	NLRI details.
[totalPaths]	*integer*	Total number of paths for this route.

2.182.1.1.1 MplsVpnRouteKeyDetail

[ipGenPrefix]	*IP address prefix*	IPv4 or IPv6 address prefix.
nlriType	*enum: IPv6 prefix | IPv4 prefix*	NLRI type.
[rd]	*string*	Route distinguisher.

2.182.1.1.2 MplsVpnRoutePath

asPathEntry	*MplsVpnRouteASPathEntry* see 2.182.1.1.2.1	AS path information.
[importedMplsVpnPathRd]	*string*	Imported MPLS VPN path RouteDistinguisher.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
[nextHop]	*string*	Route next hop address.

reasonNotBestpath	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*MplsVpnRouteDetailEntry* see 2.182.1.1.2.2	Route details.
routeType	*BgpVpnRouteTypeEntry* see 2.139.1.2.1	Route type.
[tag] **	*integer*	Tag for the route. **Since: 4.29.1
[timestamp]	*integer*	UTC seconds since epoch when the route was received.Only returned with detail output.
[weight]	*integer*	Weight for the route.

2.182.1.1.2.1 MplsVpnRouteASPathEntry

[asPath]	*string*	AS path string (if absent, then the route was originated locally).
asPathType	*enum: Invalid* | *External* | *Confed-External* | *Internal* | *Local*	AS path type: Internal - originated by I-BGP External - originated by E-BGP Confed-External - originated by a E-BGP confederation Local - originated locally Invalid - AS path is invalid.

2.182.1.1.2.2 MplsVpnRouteDetailEntry

[aggregator]	*string*	Aggregator of the route.
[clusterList]	*string*	Cluster list for the route.

communities	*list of string*	List of communities for this route.
[customerAttributeSet]	*CustomerAttributesDetail-Entry* see 2.182.1.1.2.3	Customer Attributes.
[domainPath]	*list of string*	Domain path attribute.
extCommunities	*list of string*	List of extended communities for this route.
extCommunityListRaw	*list of integer*	List of raw extended communities for this route.
fromRRClient	*boolean*	Route received from route reflector client.
[labelStack]	*list of integer*	MPLS label stack information.
largeCommunities	*list of string*	List of large communities for route.
[localLabel]	*EncapLabel* see 2.17.1.1.2.28	Local MPLS label.
[localSrv6Sid] **	*EncapLabel* see 2.17.1.1.2.28	Local SRv6 SID. **Since: 4.30.1
origin	*enum: Igp* | *Egp* | *Incomplete*	Route origin.
[originator]	*IPv4 address*	Router ID of the originator of this route.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.
[pendingTimeToAdv]	*floating point number*	Timestamp of route advertisement.
[redistributionProtocol]	*enum: Connected* | *Ospf3* | *Static* | *unknown* | *Is-Is* | *Rip* | *Ospf*	Protocol from which route got redistributed into BGP.
[remoteLabel]	*EncapLabel* see 2.17.1.1.2.28	Remote MPLS label.
[remoteSrv6Sid] **	*EncapLabel* see 2.17.1.1.2.28	Remote SRv6 SID. **Since: 4.30.1
[rxPathId]	*integer*	Received path ID of this route.
[rxSafi]	*string*	Received SAFI of this route.
[seqNumber]	*integer*	Route sequence number.
tunnelRibEligible	*boolean*	Route eligible to be installed in Tunnel RIB.
[txPathId]	*integer*	Advertised path ID for this route.

2.182.1.1.2.3 CustomerAttributesDetailEntry

[aggregator]	*string*	Aggregator of the route.
[clusterList]	*string*	Cluster list for the route.
communities	*list of string*	Standard community attributes.
[domainPath]	*list of string*	Domain path attribute.
extCommunities	*list of string*	Extended community attributes.
extCommunityListRaw	*list of integer*	Raw extended community attributes.
largeCommunities	*list of string*	Large community attributes.
[localPreference]	*integer*	I-BGP Local preference indicator.
[med]	*integer*	Multi Exit Discriminator for the route.
origin	*enum: Igp* | *Egp* | *Incomplete*	Route origin.
originAsn	*integer*	Origin AS number.

[originator]	*IPv4 address*	Router ID of the originator of this route.

2.183 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv4 ((routes | advertised-routes | (received-routes))) [...]

revision 2

see 2.182

2.184 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv6 ((routes | advertised-routes | (received-routes))) PREFIX (rd RD_VAL) internal [...]

revision 2

see 2.182

2.185 show bgp neighbors (IP_PEER | IP6_PEER | IP6_LL_PEER) vpn-ipv6 ((routes | advertised-routes | (received-routes))) [...]

revision 2

see 2.182

2.186 show bgp neighbors PEER vpls (advertised-routes | received-routes | routes) [...]

asn	*integer*	Autonomous System Number.
routerId	*IPv4 address*	BGP Router Identity.
routes	*table(string) = BgpVplsRoute* see 2.186.1	BGP VPLS routes indexed by route key.
vrf	*string*	VRF name.

2.186.1 BgpVplsRoute

paths	*list of BgpVplsPath* see 2.186.1.1	BGP VPLS route paths.
routeKeyDetail	*BgpVplsRouteKeyDetail* see 2.186.1.2	NLRI details.
totalPaths	*integer*	Total number of paths for this route.

2.186.1.1 BgpVplsPath

asPathEntry	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
[localPreference]	*integer*	Local preference.
[med]	*integer*	Multi exit discriminator.
[nextHop]	*IP address*	Next hop address.

reasonNotBestpath	*enum: ecmpFast* | *routerIdTB* | *noBest* | *ebgp* | *med* | *noLocalLabel* | *age* | *clusterlistLenTB* | *originatorId* | *macMobilitySeqNum* | *nexthopVrf* | *routeUnusable* | *aspathDetails* | *neighborMultipath* | *infMed* | *pathId* | *noReason* | *macMobilityTep-AddrdomainPathLength* | *macMobilityStickyMac* | *rrClient* | *routeNotReady* | *originAsValidity* | *igpCost* | *macMobilityRemoteDomain* | *nhgTunnelEcmpDisabled* | *aigp* | *originatorIdOrRouterId* | *origin* | *localpref* | *preference* | *clusterlistLen* | *aspathLength* | *macMobilityEsi* | *weight* | *ageTB* | *peerAddress* | *invalid* | *routerId* | *localPath* | *rtMembershipDefault-Originate* | *redist* | *originatorIdTB*	Reason route was not selected as BGP best path.
[routeDetail]	*BgpVpnRouteDetailEntry* see 2.186.1.1.1	Route details.
routeType	*BgpVpnRouteTypeEntry* see 2.139.1.2.1	Route type.
[tag] **	*integer*	Tag. **Since: 4.29.1
[timestamp]	*integer*	UTC seconds since epoch when the route was received.
[weight]	*integer*	Weight.

2.186.1.1.1 BgpVpnRouteDetailEntry

Could also be an instance of:
MplsVpnRouteDetailEntry 2.182.1.1.2.2
MvpnRouteDetailEntry 2.164.1.1.2.2
EvpnRouteDetailEntry 2.139.1.2.3

[clusterList]	*string*	Cluster list for the route.
communities	*list of string*	List of communities for this route.
extCommunities	*list of string*	List of extended communities for this route.
extCommunityListRaw	*list of integer*	List of raw extended communities for this route.
fromRRClient	*boolean*	Route received from route reflector client.
largeCommunities	*list of string*	List of large communities for route.

origin	*enum: Igp | Egp | Incomplete*	Route origin.
[originator]	*IPv4 address*	Router ID of the originator of this route.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.
[rxSafi]	*string*	Received SAFI of this route.
tunnelRibEligible	*boolean*	Route eligible to be installed in Tunnel RIB.

2.186.1.2 BgpVplsRouteKeyDetail

nlriType	*autoDiscovery*	NLRI type.
[peAddr]	*integer*	PE address.
[rd]	*string*	Route distinguisher.

2.187 show bgp neighbors [...]

revision 3

see 2.17

2.188 show bgp neighbors history [...]

vrfs	*table(string) = NeighborHistoryVrfModel* see 2.188.1	A dictionary of peer event histories keyed by VRF name.

2.188.1 NeighborHistoryVrfModel

peers	*table(IPv4 or IPv6 address) = NeighborHistoryPeerModel* see 2.188.1.1	A dictionary of peer event histories keyed by peer address.

2.188.1.1 NeighborHistoryPeerModel

events	*list of NeighborHistory-RecordModel* see 2.188.1.1.1	Peer event history records.

2.188.1.1.1 NeighborHistoryRecordModel

asn	*integer*	Peer remote AS number.
connectionType	*enum: dynamic | interface | static*	Type of peer connection.
event	*connectFailure*	Type of recorded event.
message	*string*	Event description.
time	*floating point number*	UTC timestamp of event.

2.189 show bgp neighbors history socket

vrfs	*table(string) = NeighborHistorySocketVrfModel* see 2.189.1	A dictionary of peer socket statistics histories keyed by VRF name.

2.189.1 NeighborHistorySocketVrfModel

peers	*table(IPv4 or IPv6 address) = NeighborHistorySocketPeerModel* see 2.189.1.1	A dictionary of peer socket statistics histories keyed by peer address.

2.189.1.1 NeighborHistorySocketPeerModel

localAddr	*IP address*	Peer local address.
localPort	*integer*	Peer local port.
records	*list of NeighborHistorySocketRecordModel* see 2.189.1.1.1	Peer socket statistics history records.
remotePort	*integer*	Peer remote port.

2.189.1.1.1 NeighborHistorySocketRecordModel

congestionWindow	*integer*	TCP send congestion window.
delayedAckTimeout	*integer*	TCP socket delayed ACK timeout (microseconds).
maxSegmentSize	*integer*	TCP outgoing maximum segment size.
rcvWindow	*integer*	TCP advertised receive window.
rcvWindowScale	*integer*	TCP socket receive window scale.
retransTimeout	*integer*	TCP socket retransmission timeout (microseconds).
sendWindowScale	*integer*	TCP socket send window scale.
sndRtt	*integer*	TCP round-trip time (microseconds).
sndRttVariance	*integer*	TCP round-trip time variance (microseconds).
state	*enum: CLOSING | NOTUSED | LISTEN | SYN-RECEIVED | CLOSE-WAIT | FIN-WAIT-2 | CLOSED | LAST-ACK | ESTABLISHED | TIME-WAIT | FIN-WAIT-1 | SYN-SENT*	TCP socket state.
throughput	*floating point number*	TCP throughput (megabit per second).
time	*floating point number*	UTC timestamp of record.

2.190 show bgp optimal-route-reflection position

[position]	*list of BgpOrrPosition* see 2.190.1	BGP optimal route reflection positions configured.
[vrfName]	*string*	VRF name that BGP optimal route reflection positions belong to.

2.190.1 BgpOrrPosition

active	*boolean*	The position is active.
[inactiveReason]	*string*	Reason the position is not active.
name	*string*	Name of BGP optimal route reflection position.
[numericId]	*integer*	Internal numeric identifier of position, assigned to active positions only.
[sources] **	*list of BgpOrrPositionSource* see 2.190.1.1	List of source nodes in IGP topologies. **Since: 4.27.1

2.190.1.1 BgpOrrPositionSource

instance	*integer*	Instance-id of the source.
[isisLevel]	*integer*	Isis level of the source.
protocol	*isis*	protocol advertising the source.
source	*string*	IGP source node id.

2.191 show bgp path-selection [...]

see 2.179

2.192 show bgp path-selection summary

revision 3

see 2.46

2.193 show bgp peer-group [...]

Could also be an instance of:
AllBgpPeerGroups 2.193.1.1.2

vrfs	*table(string) = BgpPeerGroups* see 2.193.1	Per-VRF BGP peer groups.

2.193.1 BgpPeerGroups

peerGroups	*table(string) = BgpPeerGroup* see 2.193.1.1	A dictionary of peer groups keyed by name.

2.193.1.1 BgpPeerGroup

[ancestorPeerGroups] **	*list of string*	Ancestor peer group list. **Since: 4.29.1
dynamicPeers	*table(string) = BgpPeerGroupEntry* see 2.193.1.1.1	A dictionary of dynamic peers keyed by IP address.
[interfacePeers] **	*table(string) = BgpPeerGroupEntry* see 2.193.1.1.1	A dictionary of interface peers keyed by IP address. **Since: 4.28.1
listenRangeSubnets	*table(string) = BgpPeerGroupListenRangeSubnet* see 2.193.1.1.2	A dictionary of listen-range subnets keyed by IP prefix.
staticPeers	*table(string) = BgpPeerGroupEntry* see 2.193.1.1.1	A dictionary of static peers keyed by IP address.

2.193.1.1.1 BgpPeerGroupEntry

negotiatedMpCapabilities	*NegotiatedMpCapabilities* see 2.193.1.1.1.1	Negotiated MP capabilities.
peerState	*enum: OpenConfirm | NotNegotiated | Active | Connect | Idle | Established | OpenSent*	Current peer state. Idle - Ignoring everything Connect - Trying to connect Active - Waiting for a connection OpenSent - Open packet has been sent OpenConfirm - Waiting for a keepalive or notify Established - Connection has been established.
vrf	*string*	VRF name.

2.193.1.1.1.1 NegotiatedMpCapabilities

dps **	*boolean*	Has Dynamic Path Selection capability. **Since: 4.24.1
ipv4FlowSpec	*boolean*	Has IPv4 flow specification capability.
[ipv4FlowSpecVpn] **	*boolean*	Has VPN-IPv4 flow specification capability. **Since: 4.30.0
ipv4MplsLabels	*boolean*	Has IPv4 labeled-unicast capability.

ipv4MplsVpn	*boolean*	Has VPN-IPv4 capability.
ipv4Multicast	*boolean*	Has IPv4 multicast capability.
[ipv4Mvpn] **	*boolean*	Has IPv4 MVPN capability. **Since: 4.28.1
ipv4SrTe	*boolean*	Has IPv4 SR-TE capability.
ipv4Unicast	*boolean*	Has IPv4 unicast capability.
ipv6FlowSpec	*boolean*	Has IPv6 flow specification capability.
[ipv6FlowSpecVpn] **	*boolean*	Has VPN-IPv6 flow specification capability. **Since: 4.30.0
ipv6MplsLabels	*boolean*	Has IPv6 labeled-unicast capability.
ipv6MplsVpn	*boolean*	Has VPN-IPv6 capability.
ipv6Multicast	*boolean*	Has IPv6 multicast capability.
ipv6SrTe	*boolean*	Has IPv6 SR-TE capability.
ipv6Unicast	*boolean*	Has IPv6 unicast capability.
l2VpnEvpn	*boolean*	Has L2VPN EVPN capability.
[l2VpnVpls] **	*boolean*	Has L2VPN VPLS capability. **Since: 4.28.1
linkState **	*boolean*	Has BGP Link State capability. **Since: 4.23.2
rtMembership **	*boolean*	Has RT Membership capability. **Since: 4.25.0

2.193.1.1.2 BgpPeerGroupListenRangeSubnet

asNumber	*string*	AS number.
peerFilter	*string*	Peer filter.
vrf	*string*	VRF name.
[emptyPeerGroups]	*list of string*	List of peer group names with no active members.
vrfs	*table(string) = BgpPeerGroups* see 2.193.1	Per-VRF BGP peer groups.

2.194 show bgp rcf [...]

functions	*table(string) = ShowAgentRcfFunctionModel* see 2.194.1	Routing control function information, keyed by function name.

2.194.1 ShowAgentRcfFunctionModel

[missingReferenceDetails] **	*ShowAgentRcfMissingRefModel* see 2.194.1.1	Detail of missing reference lists. **Since: 4.29.2
status	*enum: missingReference | missingSymbol | active*	Routing control function state. active - Routing control function is valid missingReference - Routing control function is missing an external reference missingSymbol - Routing control function symbol is missing.

2.194.1.1 ShowAgentRcfMissingRefModel

[asPathList]	*list of string*	Missing reference to an ASN list.
[communityList]	*list of string*	Missing reference to a community list.
[extCommunityList]	*list of string*	Missing reference to an extended community list.
[largeCommunityList]	*list of string*	Missing reference to a large community list.
[prefixListV4]	*list of string*	Missing reference to an IPv4 prefix list.
[prefixListV6]	*list of string*	Missing reference to an IPv6 prefix list.
[roaTable]	*list of string*	Missing reference to a ROA table.
[srLabelRangeMissing]	*boolean*	Indicate if SR label range is missing.

2.195 show bgp route-flap-damping (AFI unicast) [...]

vrfs	*table(string) = BgpRfdVrfModel* see 2.195.1	Per VRF BGP route rfd information.

2.195.1 BgpRfdVrfModel

rfdRouteEntries	*table(string) = BgpRfdRouteEntry* see 2.195.1.1	Route flap damping route entries keyed by prefix.
vrf	*string*	VRF name.

2.195.1.1 BgpRfdRouteEntry

address	*IP address*	Route address.
bgpRfdRoutePaths	*list of BgpRfdRoutePath* see 2.195.1.1.1	List of Bgp route paths.
maskLength	*integer*	Route mask length.
[totalPaths]	*integer*	Total number of paths for this route.

2.195.1.1.1 BgpRfdRoutePath

[asPathEntry]	*BgpRouteASPathEntry* see 2.17.1.1.2.1	AS path information.
dampingState	*enum: stateSuppressed | stateTracked | stateIgnored*	Route flap damping state.
[discardAt]	*floating point number*	Time until which path will be tracked.
flaps	*integer*	Number of path flaps.
nlri	*NlriModel* see 2.127.1.1	NLRI of the path.
[peerEntry]	*BgpRoutePeerEntry* see 2.17.1.1.2.35	Peer information for the route.
penalty	*integer*	Route flap damping penalty.
[reuseAt]	*floating point number*	Time at which suppressed path will be reused.
rfdPolicyName	*string*	Route flap damping policy name.
[suppressedAt]	*floating point number*	Time at which path was suppressed.
[withdrawn]	*boolean*	Path is withdrawn.

2.196 show bgp route-flap-damping policy [...]

policies	*table(string) = BgpRfdPolicy* see 2.196.1	Dictionary of route flap damping policies keyed by policy name.

2.196.1 BgpRfdPolicy

attributeChangePenalty	*integer*	Penalty for path attribute change.
maximumPenaltyThreshold	*integer*	Maximum value of penalty imposed on a path.
maximumSuppressionTime	*integer*	Maximum time a path is suppressed after it is stable in seconds.
minimumSuppressionTime	*integer*	Minimum time a path is suppressed after it is stable in seconds.
penaltyDecayHalfLife	*integer*	Penalty decay half life in seconds.
readvertisementAfter-WithdrawalPenalty	*integer*	Penalty for path readvertisement after it is withdrawn.
reusePenaltyThreshold	*integer*	Value of penalty below which suppressed path would be reused.
[suppressionIgnorePrefix-ListV4Unicast]	*string*	Prefix list of IPv4 unicast prefixes ignored from suppression.
[suppressionIgnorePrefix-ListV6Unicast]	*string*	Prefix list of IPv6 unicast prefixes ignored from suppression.
suppressionPenaltyThreshold	*integer*	Value of penalty above which path would be suppressed.
withdrawalPenalty	*integer*	Penalty for path withdrawal.

2.197 show bgp rpki roa ipv4 | ipv6 [...]

roas	*list of Roa* see 2.197.1	Resource Public Key Infrastructure (RPKI) Route Origin Authorizations (ROAs).

2.197.1 Roa

asn	*integer*	Autonomous System Number.
caches **	*list of string*	Cache servers. **Since: 4.26.2
maxLength	*integer*	Max prefix length.
prefix	*IP address prefix*	Prefix.

2.198 show bgp rpki roa summary

caches	*table(string) = CacheRoaSummaryModel* see 2.198.1	A mapping of cache names to the count of ROAs.
tables **	*table(string) = TableRoaSummaryModel* see 2.198.2	A mapping of ROA table names to the count of ROAs. **Since: 4.27.1
total	*CacheRoaSummaryModel* see 2.198.1	Total number of ROAs in the local database.

2.198.1 CacheRoaSummaryModel

ipv4RoaCount	*integer*	Count of IPv4 ROAs.
ipv6RoaCount	*integer*	Count of IPv6 ROAs.

2.198.2 TableRoaSummaryModel

ipv4RoaCount	*integer*	Count of IPv4 ROAs.
ipv6RoaCount	*integer*	Count of IPv6 ROAs.

2.199 show bgp rpki roa table summary

roaTables	*table(string) = RoaTableSummaryModel* see 2.199.1	A mapping of RPKI ROA table names to caches and status.

2.199.1 RoaTableSummaryModel

caches	*list of string*	Caches within the respective ROA table.
status	*enum: inactive | active*	State of the respective ROA table.

2.200 show bgp rt-membership [...]

see 2.180

2.201 show bgp rt-membership floodlist [...]

vrfs	*table(string) = ShowRtMembership-FloodlistVrfModel* see 2.201.1	RT membership floodlist information, keyed by VRF name.

2.201.1 ShowRtMembershipFloodlistVrfModel

routeTargets	*table(route target) = ShowRtMembership-FloodlistRouteTargetModel* see 2.201.1.1	RT membership VRF floodlist information, keyed by route target.

2.201.1.1 ShowRtMembershipFloodlistRouteTargetModel

peers	*list of IP address*	RT membership VRF and route target peer floodlist.

2.202 show bgp rt-membership origin-as (AS_PLAIN_NUM | AS_DOT_NUM) [...]

see 2.180

2.203 show bgp rt-membership route-target ROUTE_TARGET [...]

see 2.180

2.204 show bgp rt-membership summary

revision 3

see 2.46

2.205 show bgp scheduler [...]

see 2.79

2.206 show bgp segment-routing prefix-segments

`prefixSegments`	*table(IPv4 or IPv6 address prefix) = PrefixSegment* see 2.206.1	Mapping of IPv4 or IPv6 prefixes to corresponding route segments.

2.206.1 PrefixSegment

`localLabel`	*integer*	The generated local label of the route.
`sid`	*integer*	The segment ID of the route.

2.207 show bgp session tracker [...]

`[trackers]`	*table(string) = BgpSessionTrackerEntry* see 2.207.1	Dictionary of session trackers keyed by tracker name.

2.207.1 BgpSessionTrackerEntry

`establishedSessions`	*integer*	Established BGP sessions this tracker tracks.
`interfaces`	*list of Interface name*	Interfaces.
`peers`	*list of IP address*	Peer addresses.
`recoveryDelay`	*integer*	Delay for session recovery in seconds.
`remainingRecoveryDelay`	*integer*	Remaining delay for session recovery in seconds.
`state`	*enum: down | init | up | recovery*	Tracker state.
`totalSessions`	*integer*	Total BGP sessions this tracker tracks.

2.208 show bgp sr-te [...]

revision 3

see 2.17

2.209 show bgp sr-te community { (AANN_FORMAT | COMMUNITY_ENUM | COMMUNITY_U32) } [...]

revision 3

see 2.17

2.210 show bgp sr-te community-list COMM_LIST_NAME [...]

revision 3

see 2.17

2.211 show bgp sr-te endpoint ENDPOINT_VALUE color COLOR_VALUE distinguisher DISTINGUISHER_VALUE [...]

revision 3

see 2.17

2.212 show bgp sr-te endpoint ENDPOINT_VALUE path-selection color-vrf COLOR_VRF (avt AVT) router ROUTER_ID [...]

revision 3

see 2.17

2.213 show bgp sr-te summary [...]

revision 3

see 2.46

2.214 show bgp statistics [...]

[globalBgpStats] **	*BgpStatisticsGlobal* see 2.214.2	Global BGP statistics information. **Since: 4.25.2
vrfs	*table(string) = BgpStatistics* see 2.214.1	Per VRF BGP Statistics.

2.214.1 BgpStatistics

asn	*string*	Autonomous System Number.
numOfPeersByState	*PeerStateModel* see 2.214.1.1	Number of peers in each state.
routerId	*IPv4 address*	Bgp Router Identity.

2.214.1.1 PeerStateModel

[active]	*integer*	Number of peers in Active state.
[connect]	*integer*	Number of peers in connect state.
[established]	*integer*	Number of peers in Established state.
[idle]	*PeerIdleStateModel* see 2.214.1.1.1	Number of peers in idle state for different idle reasons.
[notNegotiated]	*integer*	Number of peers in NotNegotiated state.
[openConfirm]	*integer*	Number of peers in OpenConfirm state.
[openSent]	*integer*	Number of peers in OpenSent state.

2.214.1.1.1 PeerIdleStateModel

[admin]	*integer*	Number of peers in idle state because the peer is disabled.
[asuShutdown]	*integer*	Number of peers in idle state because ASU2 shutdown is in progress.
[damping] **	*integer*	Number of peers in idle state because of BFD flap damping. **Since: 4.25.1
[deleted]	*integer*	Number of peers in idle state because the peer is deleted and waiting for deffered cleanup.
[ebgpLinkLocalMultihop-Unsupported] **	*integer*	Number of EBGP ipv6 link local peers in idle state because multihop is configured. **Since: 4.22.0
[grpAdmin]	*integer*	Number of peers in idle state because the peer's group is disabled.
[ibgpLinkLocalUnsupported] **	*integer*	Number of peers in idle state because IBGP peering over IPv6 link-local address is not supported. **Since: 4.26.0
[maxAcceptedPath]	*integer*	Number of peers in idle state because the peer has exceeded its max accepted route limit.

[maxAdvertisedPath] **	*integer*	Number of peers in idle state because the peer has exceeded its max advertised route limit. **Since: 4.24.0
[maxPath]	*integer*	Number of peers in idle state because the peer has exceeded its max route limit.
[minHoldTime] **	*integer*	Number of peers in idle state because peer hold time received in open message is less than minimum hold time required. **Since: 4.29.0
[noActivate]	*integer*	Number of peers in idle state because the peer is not activated in any address-family mode.
[noIf]	*integer*	Number of peers in idle state because the peer's interface is down.
[noLocalNexthops]	*integer*	Number of peers in idle state because the peer is missing local addresses for activated AFI/SAFIs.
[noRemoteAs]	*integer*	Number of peers in idle state because remote-as for peer is missing.
[orrAndAddPathBackup] **	*integer*	Number of peers in idle state because they are configured with both ORR and add-path backup, they are not supported together. **Since: 4.27.2
[orrLinkLocalPeer] **	*integer*	Number of peers in idle state because they are link-local peers configured with ORR that is not supported. **Since: 4.26.2
[orrPositionCountExceeds-Maximum] **	*integer*	Number of peers in idle state because they are configured with an ORR position we could not assign a numeric ID to, as the total number of configured ORR positions exceeds the maximum number supported. **Since: 4.26.2
[orrPositionIncomplete] **	*integer*	Number of peers in idle state because they are configured with ORR positions that either does not exist, or does not have at least one IGP prefix configured. **Since: 4.26.2
[orrPositionPeerAddress-DynamicPeer] **	*integer*	Number of peers in idle state because they are dynamic peers configured with ORR peer-address position that is not supported. **Since: 4.26.2
[other]	*integer*	Number of peers in idle state because of some unknown reason.
[pendingNotification] **	*integer*	Number of peers in idle state because the peer is pending notification message to be sent. **Since: 4.21.0
[unsupportedCaps]	*integer*	Number of peers in idle state because the peer has not negotiated required capabilities.

2.214.2 BgpStatisticsGlobal

`policyAcceptedPaths`	*integer*	Number of policy accepted paths in BGP.

2.215 show bgp summary [...]

vrfs	*table(string) = BgpSummaryAll* see 2.215.1	Per VRF BGP summary.

2.215.1 BgpSummaryAll

asn	*string*	Autonomous System Number.
peers	*table(string) = BgpSummaryAllPeerEntry* see 2.215.1.1	Dictionary of BGP peer entries indexed by the peer address.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF Name.

2.215.1.1 BgpSummaryAllPeerEntry

[dps]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for Dynamic Path Selection.
[ipv4FlowSpec]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 flow specification.
[ipv4FlowSpecVpn] **	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for VPN-IPv4 flow specification. **Since: 4.30.0
[ipv4MplsLabels]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 labeled-unicast.
[ipv4MplsVpn]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for VPN-IPv4.
[ipv4Multicast]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 multicast.
[ipv4Mvpn] **	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 MVPN. **Since: 4.28.2
[ipv4SrTe]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 SR-TE.
[ipv4Unicast]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv4 unicast.
[ipv6FlowSpec]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv6 flow specification.
[ipv6FlowSpecVpn] **	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for VPN-IPv6 flow specification. **Since: 4.30.0
[ipv6MplsLabels]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv6 labeled-unicast.

[ipv6MplsVpn]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for VPN-IPv6.
[ipv6Multicast]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv6 multicast.
[ipv6SrTe]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv6 SR-TE.
[ipv6Unicast]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for IPv6 unicast.
[l2VpnEvpn]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for L2VPN EVPN.
[l2VpnVpls] **	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for L2VPN VPLS. **Since: 4.28.0
[linkState]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for BGP Link State.
peerAsn	*string*	Autonomous System Number.
peerState	*enum: OpenConfirm | NotNegotiated | Active | Connect | Idle | Established | OpenSent*	Current peer state. Idle - Ignoring everything Connect - Trying to connect Active - Waiting for a connection OpenSent - Open packet has been sent OpenConfirm - Waiting for a keepalive or notify Established - Connection has been established.
[rtMembership]	*BgpSummaryAllPeerAfiSafiEntry* see 2.215.1.1.1	Peer information for RT Membership.

2.215.1.1.1 BgpSummaryAllPeerAfiSafiEntry

afiSafiState	*enum: advertised | configured | negotiated | received*	current afiSafi state.
nlrisAccepted	*integer*	Number of nlris accepted from the peer.
nlrisReceived	*integer*	Number of nlris receieved from the peer.

2.216 show bgp traffic-policy field-set mappings [...]

vrfs	*table(string) = ShowTrafficPolicyFieldSetCommunityMappingsModel* see 2.216.1	Per VRF BGP traffic-policy field-set mappings.

2.216.1 ShowTrafficPolicyFieldSetCommunityMappingsModel

vrfs	*table(string) = TrafficPolicyFieldSetCommunityMappingsVrfModel* see 2.216.1.1	A dictionary of traffic-policy field-set mappings keyed by VRF name.

2.216.1.1 TrafficPolicyFieldSetCommunityMappingsVrfModel

fieldSetMapping	*boolean*	Dynamic mapping of IP prefixes to field sets is enabled.
[ipv4Bindings]	*table(string) = TrafficPolicyFieldSetBindings* see 2.216.1.1.1	Dictionary of IPv4 prefix field set bindings indexed by the field set name.
[ipv6Bindings]	*table(string) = TrafficPolicyFieldSetBindings* see 2.216.1.1.1	Dictionary of IPv6 prefix field set bindings indexed by the field set name.
vrf	*string*	VRF name.

2.216.1.1.1 TrafficPolicyFieldSetBindings

[communities]	*list of string*	Associated standard BGP communities.
[peerGroups] **	*list of string*	Peer groups included in field-set. **Since: 4.29.2
[peerType] **	*enum: ebgp | ibgp*	Type of peers included in field-set. **Since: 4.29.2

2.217 show bgp update-group [...]

vrfs	*table(string) = BgpUpdateGroups* see 2.217.1	Per VRF BGP update group.

2.217.1 BgpUpdateGroups

updateGroups	*table(numeric string) = BgpUpdateGroup* see 2.217.1.1	Dictionary of Bgp update groups by update group index.

2.217.1.1 BgpUpdateGroup

advRoutes	*integer*	Number of advertised routes.
[apSendCap] **	*boolean*	Additional path send capability. **Since: 4.24.0
[apSendV4UniCap]	*boolean*	IPv4 Unicast additional path send capability.
[apSendV6UniCap]	*boolean*	IPv6 Unicast additional path send capability.
[confedExternalPeer] **	*boolean*	eBGP confederation peer. **Since: 4.24.0
[defaultMed] **	*integer*	Default MED. **Since: 4.24.0
[defaultOriginate] **	*enum: always | enabled*	Advertise a default route to the neighbor. **Since: 4.24.0
[exportLocalPref]	*integer*	Local preference to send.
[extendedNextHop] **	*enum: disabled | enabled | enabledWithOriginate*	Extended next hop capability. **Since: 4.24.0
[localAs]	*integer*	Local as number.
membersInSync	*integer*	Number of in-sync members.
membersTotal	*integer*	Number of members.
[metricOut]	*integer*	Metric to send.
[missingPolicyAction] **	*enum: permit | deny*	Missing policy action. **Since: 4.24.0
[mpBgpAfiSafiCap] **	*enum: l2VpnVpls | ipv6FlowSpec | linkState | ipv6SrTe | l2VpnEvpn | ipv6MplsVpn | ipv6FlowSpecVpn | ipv4FlowSpecVpn | ipv4Unicast | dps | ipv6Multicast | ipv6Unicast | rtMembership | ipv4MplsVpn | ipv4FlowSpec | ipv6MplsLabels | ipv4Mvpn | ipv4SrTe | ipv4Multicast | ipv4MplsLabels*	MPBGP AFI SAFI capability. **Since: 4.24.0
[mpBgpV4LabelCap]	*boolean*	MPBGP capability IPv4 Label Unicast.
[mpBgpV4UniCap]	*boolean*	MPBGP capability IPv4 Unicast.
[mpBgpV4Vpn4Cap]	*boolean*	MPBGP capability IPv4 VPN4.

[mpBgpV6LabelCap] **	*boolean*	MPBGP capability IPv6 Label Unicast. **Since: 4.21.3
[mpBgpV6UniCap]	*boolean*	MPBGP capability IPv6 Unicast.
[negotiated4byteAs] **	*boolean*	Four-byte ASN negotiated. **Since: 4.24.0
[neighborAs] **	*integer*	Neighbor AS number. **Since: 4.24.0
[nexthopSelf]	*boolean*	Export self as nexthop.
[nexthopUnchanged] **	*boolean*	Preserve original next hop while advertising routes to eBGP peers. **Since: 4.24.0
[orrPositionId] **	*integer*	ORR position ID. **Since: 4.27.2
[orrPositionName] **	*string*	ORR position name. **Since: 4.27.2
[outDelay]	*integer*	Out delay time.
[outDelayApply] **	*enum: newRoutes | newRoutesChangesWithdrawals | newRoutesChanges*	Announcment types to which out delay is applied. newRoutes - Out delay is applied only to new routes newRoutesChanges - Out delay is applied to to new routes and changes newRoutesChangesWithdrawals - Out delay is applied to new routes, changes, and withdrawals. **Since: 4.21.3
[outdelayLbw]	*integer*	Out delay time for linkband width.
peerType	*enum: external | internal*	Peer type.
peers	*table(string) = BgpUpdateGroupPeerEntry* see 2.217.1.1.1	Dictionary of BGP Update Group entries indexed by peer address.
[policyGroupId] **	*integer*	RibOut policy group ID. **Since: 4.27.2
[prefixList] **	*string*	Prefix list (present if configured in ribd mode, only present if configured and defined in multi-agent mode). **Since: 4.24.0
[prefixListV4Multicast] **	*string*	Prefix List for IPv4 Multicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode). **Since: 4.29.1
[prefixListV4Uni]	*string*	Prefix List for IPv4 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[prefixListV6Multicast] **	*string*	Prefix List for IPv6 Multicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode). **Since: 4.29.1
[prefixListV6Uni]	*string*	Prefix List for IPv6 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).

[prependOwnDisabled] **	*boolean*	Do not prepend own AS number. **Since: 4.22.0
queuedAdv	*integer*	Number of queued route advertisements.
[reflectedRoutesAttributes-State] **	*enum: preserved* | *notPreserved* | *alwaysPreserved*	State of attributes of the reflected routes. **Since: 4.24.0
[removePrivateAs]	*boolean*	Remove private as numbers.
[removePrivateAsAlways] **	*boolean*	Always remove private AS numbers. **Since: 4.24.0
[removePrivateAsReplace] **	*boolean*	Replace private AS number with local AS number. **Since: 4.24.0
[replaceAs] **	*integer*	Replace AS number. **Since: 4.24.0
[routeMapOut]	*string*	Outbound policy (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutDefOri]	*string*	Outbound policy of default originate (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutDefOriV4Uni]	*string*	Outbound policy of default originate for IPv4 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutDefOriV6Uni]	*string*	Outbound policy of default originate for IPv6 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutMaint]	*string*	Outbound policy for maintenance mode (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutV4Uni]	*string*	Outbound policy for IPv4 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutV6Uni]	*string*	Outbound policy for IPv6 Unicast (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeMapOutVpn4]	*string*	Outbound policy for VPN4 (present if configured in ribd mode, only present if configured and defined in multi-agent mode).
[routeRefreshCap]	*boolean*	Route refresh capability.
[rrClient]	*boolean*	Route Reflector Client.
[rrClusterId] **	*IPv4 address*	Cluster ID assigned to route reflector client. **Since: 4.26.2
[rtMembershipConstrain] **	*boolean*	Constrain route distribution based on peer RT membership. **Since: 4.24.0

[sendCommunity]	*boolean*	Send community attribute.
[sendExtendedCommunity] **	*boolean*	Send extended community attribute. **Since: 4.24.0
[sendLargeCommunity] **	*boolean*	Send large community attribute. **Since: 4.24.0
[sendOriginAsValidity] **	*boolean*	Send origin AS validation state. **Since: 4.24.0
[sixPe] **	*boolean*	6PE activated for the neighbor. **Since: 4.24.0

2.217.1.1.1 BgpUpdateGroupPeerEntry

[inSync]	*boolean*	Peer is in-sync.
[queuedAdv]	*integer*	Number of queued advertisements.

2.218 show bgp vpls [...]

see 2.186

2.219 show bgp vpls auto-discovery [...]

instances	*table(string) = BgpVplsAdInstance* see 2.219.1	BGP VPLS auto discovered instances indexed by instance name.

2.219.1 BgpVplsAdInstance

groups	*table(string) = BgpVplsAdGroup* see 2.219.1.1	BGP VPLS auto discovered groups indexed by group name.

2.219.1.1 BgpVplsAdGroup

exportRts	*list of integer*	Route targets to export.
importRts	*list of integer*	Route targets to import.
l2VpnId	*integer*	L2 VPN identifier.
[localPeAddr]	*integer*	Local PE address.
rd	*string*	VPLS group route distinguisher.
remotePeers	*table(string) = BgpVplsAdGroupMember* see 2.219.1.1.1	Remote peers indexed by PE address.

2.219.1.1.1 BgpVplsAdGroupMember

nextHop	*IP address*	Remote PE next hop.

2.220 show bgp vpls summary

revision 3

see 2.46

2.221 show bgp vpn-ipv4 PREFIX (rd RD_VAL) internal

revision 2

see 2.182

2.222 show bgp vpn-ipv4 [...]

revision 2

see 2.182

2.223 show bgp vpn-ipv6 PREFIX (rd RD_VAL) internal

revision 2

see 2.182

2.224 show bgp vpn-ipv6 [...]

revision 2

see 2.182

2.225 show boot stages log

`[logs]`	*list of BootStagesLogEntryModel* see 2.225.1	List of boot stage logs.

2.225.1 BootStagesLogEntryModel

`deltaTime`	*floating point number*	Time of the log relative to stage start.
`msg`	*string*	Log message.
`timestamp`	*floating point number*	Timestamp when the log event occured.

2.226 show boot-config

abootPassword	*string*	Encrypted aboot password.
aristaCert **	*string*	Arista certificate. **Since: 4.22.0
aristaCert2 **	*string*	Arista certificate #2. **Since: 4.29.0
aristaCert3 **	*string*	Arista certificate #3. **Since: 4.29.0
aristaCertEnabled **	*boolean*	SWIs signed by Arista Networks are authorized to boot. **Since: 4.22.0
certsLoaded **	*boolean*	Certificates were successfully loaded. **Since: 4.22.0
[consoleSpeed]	*integer*	Baud rate of console.
fileChecksum **	*string*	Checksum of the underlying boot-config file. **Since: 4.28.1
fileChecksumAlg **	*string*	Algorithm used to hash the underlying boot-config file. **Since: 4.28.1
measuredbootEnabled **	*boolean*	Measuring boot components via TPM PCRs is enabled. **Since: 4.27.0
memTestIterations	*integer*	Iterations of Memtest to perform at boot time.
[personalityOnReboot] **	*enum: bmc | cpu*	The personality on reboot. **Since: 4.22.1
securebootEnabled **	*boolean*	Secure boot is enabled. **Since: 4.22.0
securebootSupported **	*boolean*	Secure boot is supported. **Since: 4.22.0
softwareImage	*string*	Location of software image.
spiFlashWriteProtected **	*boolean*	SPI flash is physically protected against writes. **Since: 4.22.0
spiUpdateEnabled **	*boolean*	Updating the SPI flash from Aboot is supported and enabled. **Since: 4.26.0
tpmPassword **	*boolean*	Aboot password is stored in the TPM. **Since: 4.22.0
upgradeCert **	*string*	SPI upgrade certificate. **Since: 4.27.0
userCert **	*string*	User certficate. **Since: 4.22.0

2.227 show boot-extensions

extensions	*list of string*	List of extensions that will be installed and run at startup.

2.228 show bridge ((mac address-table) | mac-address-table) aging time-out

see 2.27

2.229 show cdp neighbors

cdpInterfaces	*table(Interface name) = CdpInterface* see 2.229.1	CDP enabled interfaces.
[lastRxTime]	*floating point number*	The last time CDP Agent received packet.
nbrDeletes	*integer*	The total numer of removed neighbors.
nbrInserts	*integer*	The total numer of added neighbors.
rxEnabled **	*boolean*	True if CDP packets are parsed. **Since: 4.30.1
rxFrameErrorsTotal	*integer*	The total numer of frames received by the CDP Agent that were discarded because of invalid frame.
rxFramesTotal	*integer*	The total number of packets that CDP Agent received.
rxFramesValidTotal	*integer*	The total number of valid frames received by the CDP Agent.
rxProtocolErrorsTotal	*integer*	The total numer of frames received by the CDP Agent that were discarded because of errors in CDP protocol decoding.
rxTLVsDiscardedTotal	*integer*	The total numer of discarded TLVs received by the CDP Agent.
rxTLVsUnrecognizedTotal	*integer*	The total numer of unknown TLVs received by the CDP Agent.

2.229.1 CdpInterface

cdpNeighbors	*list of CdpNeighbor* see 2.229.1.1	CDP neighbors.

2.229.1.1 CdpNeighbor

localPort	*interface name*	Name of the interface.
[neighborDevice]	*string*	Name of the neighbor device.
[neighborPort]	*string*	Name of the neighbor port.
ttl	*integer*	The Time To Live value (in seconds) advertised by the neighbor.

2.230 show cfm alarm indication [...]

domains	*table(string) = MdAisModel* see 2.230.1	A mapping between MD name and MD AIS.

2.230.1 MdAisModel

associations	*table(string) = MaAisModel* see 2.230.1.1	A mapping between MA name and MA AIS.
level	*integer*	MD level.

2.230.1.1 MaAisModel

direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*table(numeric string) = LocalMepAisModel* see 2.230.1.1.1	Local MEPs, keyed by local MEP ID.

2.230.1.1.1 LocalMepAisModel

aisCondition	*AisDefectTimeChangeModel* see 2.230.1.1.1.1	Local MEP AIS defect condition and its last change timestamp.
aisDefects	*table(string) = AisDefectTimeChangeModel* see 2.230.1.1.1.1	AIS defects keyed by their name.
[aisTxInterval]	*integer*	AIS transmission interval in seconds.
[clientDomainLevel]	*integer*	Client MD level for AIS transmission.
[clientDomainLevelError]	*enum: notConfigured | invalid*	Reason for client domain level configuration error.
intf	*interface name*	Local maintenance end point interface.
remoteAisRxEntries	*list of RemoteAisRxEntryModel* see 2.230.1.1.1.2	List of AIS information received from remote MEPs in order of last received.

2.230.1.1.1.1 AisDefectTimeChangeModel

detected	*boolean*	End point has an AIS defect.
lastClearedTime	*floating point number*	UTC time of last AIS state cleared.
lastDetectedTime	*floating point number*	UTC time of last AIS state detected.

2.230.1.1.1.2 RemoteAisRxEntryModel

count	*integer*	Number of AIS PDUs that were received.
lastReceivedTime	*floating point number*	UTC time of last received AIS PDU.
remoteMac	*MAC address*	MAC address of remote MEP that detected a fault.
txInterval	*integer*	AIS transmission interval in seconds.

2.231 show cfm continuity-check end-point [...]

domains	*table(string) = MdCcModel* see 2.231.1	A mapping between MD name and MD continuity check.

2.231.1 MdCcModel

associations	*table(string) = MaCcModel* see 2.231.1.1	A mapping between MA name and MA continuity check.
level **	*integer*	MD level. **Since: 4.28.1

2.231.1.1 MaCcModel

ccmTxInterval	*floating point number*	Continuity check transmission interval in seconds.
direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*table(numeric string) = LocalMepCcModel* see 2.231.1.1.1	Local MEPs, keyed by local MEP ID.

2.231.1.1.1 LocalMepCcModel

[ccmDefects] **	*list of CcmDefectModel* see 2.231.1.1.1.1	CCM defects. **Since: 4.29.1
intf	*interface name*	Local maintenance end point interface.
rdiTxCondition **	*RdiAndTimeChangeModel* see 2.231.1.1.1.2	Local MEP Tx RDI Bit status and its last change timestamp. **Since: 4.29.1
remoteMeps	*table(numeric string) = RemoteMepCcModel* see 2.231.1.1.1.3	Remote MEPs, keyed by remote MEP ID.

2.231.1.1.1.1 CcmDefectModel

cfmDefectPkts	*integer*	Number of defective CFM packets received on the interface.
defectId	*enum: defectCodeMdlMismatch | defectCodeCcmNotReceived | defectCodeUnconfiguredMep | defectCodeRdiRecieved | defectCodeNone | defectCodeDuplicateMep | defectCodeMaidMismatch | defectCodeCcmTxMismatch*	Defect ID.
firstSeenTime	*floating point number*	UTC time of first time the defect was detected.
lastSeenTime	*floating point number*	UTC time of last time the defect was detected.

level	*integer*	MD level.
maName	*string*	Maintenance association.
remoteMepId	*integer*	Remote MEP ID.

2.231.1.1.1.2 RdiAndTimeChangeModel

lastClearedTime	*floating point number*	UTC time of last RDI state cleared.
lastDetectedTime	*floating point number*	UTC time of last RDI state detected.
rdi	*boolean*	End point remote defect indication.

2.231.1.1.1.3 RemoteMepCcModel

locState	*LocAndTimeChangeModel* see 2.231.1.1.1.4	End point connectivity state and its last change timestamp.
[mismatchCcmTxInterval] **	*floating point number*	Mismatching Continuity check transmission interval from remote MEP. **Since: 4.29.2

2.231.1.1.1.4 LocAndTimeChangeModel

lastClearedTime	*floating point number*	UTC time of last LOC state cleared.
lastDetectedTime	*floating point number*	UTC time of last LOC state detected.
loc	*boolean*	End point connectivity state.

2.232 show cfm continuity-check end-point remote-defect [...]

domains	*table(string) = MdRdiModel* see 2.232.1	A mapping between MD name and MD RDI.

2.232.1 MdRdiModel

associations	*table(string) = MaRdiModel* see 2.232.1.1	A mapping between MA name and MA RDI.
level **	*integer*	MD level. **Since: 4.28.1

2.232.1.1 MaRdiModel

direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*table(numeric string) = LocalMepRdiModel* see 2.232.1.1.1	Local MEPs, keyed by local MEP ID.

2.232.1.1.1 LocalMepRdiModel

intf	*interface name*	Local maintenance end point interface.
rdiCondition	*RdiAndTimeChangeModel* see 2.231.1.1.1.2	Local MEP remote defect condition and its last change timestamp.
rdiTxCondition **	*RdiAndTimeChangeModel* see 2.231.1.1.1.2	Local MEP Tx RDI bit status and its last change timestamp. **Since: 4.29.1
remoteMeps	*table(numeric string) = RemoteMepRdiModel* see 2.232.1.1.1.1	Remote MEPs, keyed by remote MEP ID.

2.232.1.1.1.1 RemoteMepRdiModel

rdiState	*RdiAndTimeChangeModel* see 2.231.1.1.1.2	End point remote defect state and its last change timestamp.

2.233 show cfm counters [...]

interfaces	*table(Interface name) = CfmIntfCounter* see 2.233.1	Maps CFM counters to the corresponding interfaces.

2.233.1 CfmIntfCounter

errorPkts	*integer*	Number of invalid CFM packets received on the interface.
inPkts	*integer*	Number of CFM packets received on the interface.
outPkts	*integer*	Number of CFM packets sent from the interface.

2.234 show cfm end-point [...]

domains	*table(string) = MdModel* see 2.234.1	A mapping between MD name and MD.

2.234.1 MdModel

associations	*table(string) = MaModel* see 2.234.1.1	A mapping between MA name and MA.
level **	*integer*	MD level. **Since: 4.28.1

2.234.1.1 MaModel

ccmTxInterval	*floating point number*	Continuity check transmission interval in seconds.
direction	*enum: unconfigured | down | up*	MEP direction.

localMeps	*table(numeric string) = LocalMepModel* see 2.234.1.1.1	Local MEPs, keyed by local MEP ID.

2.234.1.1.1 LocalMepModel

intf	*interface name*	Local maintenance end point interface.
remoteMeps	*table(numeric string) = RemoteMepModel* see 2.234.1.1.1.2	Remote MEPs, keyed by remote MEP ID.
status	*enum: inactive | active*	Local maintenance end point status.
[vlanStatus] **	*LocalMepVlanModel* see 2.234.1.1.1.1	Local maintenance end point VLAN information. **Since: 4.30.1

2.234.1.1.1.1 LocalMepVlanModel

[activeVlans]	*list of integer*	Active VLANs.
[inactiveVlans]	*list of integer*	Inactive VLANs.
[primaryVlan]	*integer*	Primary VLAN.

2.234.1.1.1.2 RemoteMepModel

[reason]	*string*	Reason for remote MEP status.
status	*enum: rdi | inactive | unreachable | active*	Remote maintenance end point status.

2.235 show cfm intermediate-point

intermediatePoint	*table(string) = boolean*	Maps maintenance domains to intermediate point.

2.236 show cfm measurement delay proactive [...]

domains	*table(string) = MdDmModel* see 2.236.1	A mapping between MD name and MD.

2.236.1 MdDmModel

associations	*table(string) = MaDmModel* see 2.236.1.1	A mapping between MA name and MA.
level	*integer*	MD level.

2.236.1.1 MaDmModel

direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*table(numeric string) = LocalMepDmModel* see 2.236.1.1.1	Local MEPs, keyed by local MEP ID.
operMode	*enum: on-demand | proactive*	Delay measurement type.
txInterval	*floating point number*	Delay measurement frame transmission interval in milliseconds.

2.236.1.1.1 LocalMepDmModel

intf	*interface name*	Local maintenance end point interface.
remoteMeps	*table(numeric string) = RemoteMepDmModel* see 2.236.1.1.1.1	Remote MEPs, keyed by remote MEP ID.

2.236.1.1.1.1 RemoteMepDmModel

[avgDelay]	*floating point number*	Average measured delay.
[avgJitter]	*floating point number*	Average measured jitter.
[bestDelay]	*floating point number*	Best measured delay.
[bestDelayTime]	*floating point number*	UTC time when best delay was observed.
[delayHistory]	*list of DmStatsModel* see 2.236.1.1.1.2	Measured delay in the recent delay measurement frames.
enabled	*boolean*	Delay measurement is enabled.
[numSamples]	*integer*	Number of delay measurement frames.
[reason]	*enum: notConfigured | inProgress | failed*	Reason for delay measurement failure.
[startTime]	*floating point number*	UTC start time of delay measurement.
[worstDelay]	*floating point number*	Worst measured delay.
[worstDelayTime]	*floating point number*	UTC time when worst delay was observed.

2.236.1.1.1.2 DmStatsModel

`delay`	*floating point number*	Measured delay in this delay measurement frame.
`updateTime`	*floating point number*	UTC time when this DM stat entry was updated.

2.237 show cfm measurement loss proactive [...]

domains	*table(string) = MdLmModel* see 2.237.1	A mapping between MD name and MD.

2.237.1 MdLmModel

associations	*table(string) = MaLmModel* see 2.237.1.1	A mapping between MA name and MA.
level	*integer*	MD level.

2.237.1.1 MaLmModel

direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*table(numeric string) = LocalMepLmModel* see 2.237.1.1.1	Local MEPs, keyed by local MEP ID.
operMode	*enum: on-demand | proactive*	Loss measurement type.
txInterval	*floating point number*	Loss measurement frame transmission interval in milliseconds.

2.237.1.1.1 LocalMepLmModel

intf	*interface name*	Local maintenance end point interface.
remoteMeps	*table(numeric string) = RemoteMepLmModel* see 2.237.1.1.1.1	Remote MEPs, keyed by remote MEP ID.

2.237.1.1.1.1 RemoteMepLmModel

[avgFarEndFrameLoss]	*integer*	Average far-end frame loss.
[avgNearEndFrameLoss]	*integer*	Average near-end frame loss.
enabled	*boolean*	Loss measurement is enabled.
[lmHistory]	*list of LmStatsModel* see 2.237.1.1.1.2	Measurement stats from recent loss measurement frames.
[numSamples]	*integer*	Number of loss measurement frames.
[reason]	*enum: notConfigured | inProgress | failed*	Reason for loss measurement failure.
[startTime]	*floating point number*	UTC start time of loss measurement.

2.237.1.1.1.2 LmStatsModel

farEndLoss	*integer*	Packets lost on remote MEP.
localRx	*integer*	Packets received on local MEP.
localTx	*integer*	Packets sent from local MEP.
nearEndLoss	*integer*	Packets lost on local MEP.
remoteRx	*integer*	Packets received on remote MEP.
remoteTx	*integer*	Packets sent from remote MEP.

`updateTime`	*floating point number*	UTC time when this LM stat entry was updated.

2.238 show cfm measurement loss synthetic proactive [...]

domains	*table(string) = MdSlmModel* see 2.238.1	A mapping between MD name and MD.

2.238.1 MdSlmModel

associations	*table(string) = MaSlmModel* see 2.238.1.1	A mapping between MA name and MA.
level	*integer*	MD level.

2.238.1.1 MaSlmModel

direction	*enum: unconfigured | down | up*	MEP direction.
localMeps	*list of LocalMepSlmModel* see 2.238.1.1.1	List of Local MEPs and their COS value.
operMode	*enum: on-demand | proactive*	Synthetic Loss measurement type.
txInterval	*floating point number*	Synthetic Loss measurement frame transmission interval in milliseconds.

2.238.1.1.1 LocalMepSlmModel

cos	*integer*	COS value used for Synthetic Loss measurement frames.
intf	*interface name*	Local MEP interface.
mepId	*integer*	Local MEP ID.
remoteMeps	*table(numeric string) = RemoteMepSlmModel* see 2.238.1.1.1.1	Remote MEPs, keyed by remote MEP ID.

2.238.1.1.1.1 RemoteMepSlmModel

[avgFarEndFrameLoss]	*floating point number*	Average far-end synthetic frame loss.
[avgNearEndFrameLoss]	*floating point number*	Average near-end synthetic frame loss.
enabled	*boolean*	Synthetic loss measurement is enabled.
[measurementPeriod]	*floating point number*	Synthetic loss measurement period in milliseconds.
[numMeasurements]	*integer*	Number of measurements taken.
[reason]	*enum: notConfigured | inProgress | failed*	Reason for synthetic loss measurement failure.
[slmHistory]	*list of SlmStatsModel* see 2.238.1.1.1.2	Measurement stats from recent synthetic loss measurement frames.
[startTime]	*floating point number*	UTC start time of synthetic loss measurement.

2.238.1.1.1.2 SlmStatsModel

localRx	*integer*	Packets received on local MEP.
localTx	*integer*	Packets sent from local MEP.
remoteRx	*integer*	Packets received on remote MEP.
updateTime	*floating point number*	UTC time of the measurement period.

2.239 show class-map [...]

classMaps	*table(string) = ClassMapModel* see 2.239.1	List of all class maps.

2.239.1 ClassMapModel

classPrio	*integer*	Class priority.
[match]	*table(string) = Match* see 2.239.1.1	Matching access groups.
matchCondition	*enum: unspecified | matchConditionAny*	Match condition.

2.239.1.1 Match

[dscpValue] **	*integer*	DSCP equivalent value in digits. **Since: 4.20.0
[l2ParamsValue] **	*MatchL2Params* see 2.239.1.1.1	L2 Params values configured. **Since: 4.23.2
[matchDscpNameValid] **	*boolean*	Flag to identify if string value of DSCP is used or numeric value. **Since: 4.20.0
[matchEcnValue] **	*enum: non-ect | ce | ect | dontCare | ect-ce*	ECN value configured. **Since: 4.20.0
matchRule	*enum: matchIpAccessGroup | matchDscpEcn | matchMacAccessGroup | matchL2Params | matchMplsTrafficClass | matchIpv6AccessGroup*	Matching rule for the type.

2.239.1.1.1 MatchL2Params

[cosValue]	*string*	COS value configured.
[deiValue] **	*string*	DEI value configured. **Since: 4.29.0
[innerCosValue] **	*string*	Inner COS value configured. **Since: 4.26.2
[innerVlanMask]	*integer*	Inner VLAN mask used.
[innerVlanMaskValid]	*boolean*	Flag to identify if inner vlan range type is used or (innervlan id, mask) is used.
[innerVlanValue]	*string*	Inner VLAN match value configured.
[vlanMask]	*integer*	VLAN mask used.
[vlanMaskValid]	*boolean*	Flag to identify if vlan range type is used or (vlan id, mask).
[vlanValue]	*string*	VLAN match value configured.

2.240 show class-map type control-plane [...]

see 2.239

2.241 show class-map type pbr [...]

classMaps	*table(string) = ClassMapModel* see 2.241.1	List of all class maps.
mapType	*enum: mapTrafficPolicy | mapSamplePolicy | mapServerLoadBalancer | mapQos | mapTapAgg | mapPbr | mapFlowspec*	Map type.

2.241.1 ClassMapModel

[counterData]	*ClassMapCounter* see 2.241.1.1	Number of packets that matched this class-map.
[match]	*table(string) = ClassMatchModel* see 2.241.1.2	Matching rule for class map.
matchCondition	*enum: matchConditionAny | matchConditionAll*	Class map match condition.
name	*string*	Class map name.

2.241.1.1 ClassMapCounter

[checkpointPacketCount]	*integer*	Number of packets matching the rule since last checkpoint.
[lastChangedTime]	*floating point number*	Last time the counters changed.
[packetCount]	*integer*	Number of packets matching the rule.

2.241.1.2 ClassMatchModel

[acl]	*table(numeric string) = ClassMatchAclModel* see 2.241.1.2.1	Matching acl rule for class map.
[ip6Rule]	*table(numeric string) = Ipv6Rule* see 2.241.1.2.3	Matching IPv6 rule config.
[ipRule]	*table(numeric string) = IpRule* see 2.241.1.2.2	Matching IP rule config.
[macRule]	*table(numeric string) = MacRule* see 2.241.1.2.4	Matching MAC rule config.
[mplsRule]	*MplsRule* see 2.241.1.2.5	Matching MPLS rule config.
option	*enum: matchIpAccessGroup | matchIpv6AccessGroup | matchMplsAccessGroup | matchMacAccessGroup*	class match protocol options.

2.241.1.2.1 ClassMatchAclModel

[ipAclList]	*IpAclList* see 2.53.1	IP Acl list.
[ipv6AclList]	*Ipv6AclList* see 2.53.2	IPv6 Acl list.
[macAclList]	*MacAclList* see 2.241.1.2.1.1	MAC Acl list.
name	*string*	acl name.

2.241.1.2.1.1 MacAclList

aclList	*list of _AclBase* see 2.53.1.1	List of ACLs.

2.241.1.2.2 IpRule

[action]	*enum: permit | deny*	Action to take when the filter matches.
convertSymbols **	*boolean*	Auto-conversion from subnet and port numbers to words in show commands. **Since: 4.20.5
[copyToCpuDst] **	*enum: captive-portal | none*	Copy packet to CPU when this rule is matched. **Since: 4.25.1
[counterData]	*AclCounter* see 2.241.1.2.2.1	counter data for packets matching this rule.
[log]	*boolean*	Log when this rule is matched.
[mirrorSession] **	*string*	Target mirror session if this rule is matched. **Since: 4.23.0
[payload]	*PayloadSpec* see 2.52.1	DPI based payload filter specification.
[ruleFilter]	*IpFilter* see 2.241.1.2.2.2	The filter criteria to match this rule. Unset for remarks. When an optional filter attribute is not present in the output, this implies that there was no specification configured for that attribute, and any valid value is permitted.
sequenceNumber	*integer*	Numeric sequence number.
text	*string*	A textual representation of the rule, sans sequence number. If this rule is a remark, this field will contain the remark text.

2.241.1.2.2.1 AclCounter

[byteCount] **	*integer*	Number of bytes matching the rule. **Since: 4.29.1
[checkpointByteCount] **	*integer*	Number of bytes matching the rule since last checkpoint. **Since: 4.29.1
[checkpointConnCount] **	*integer*	Number of connections matching the rule since last checkpoint. **Since: 4.20.0

[checkpointPacketCount]	*integer*	Number of packets matching the rule since last checkpoint.
[connCount] **	*integer*	Number of connections matching the rule. **Since: 4.20.0
[lastChangedTime]	*floating point number*	Last time the counters changed.
[packetCount]	*integer*	Number of packets matching the rule.

2.241.1.2.2.2 IpFilter

destination	*IpAddrAndMask* see 2.241.1.2.2.9	Destination IPv4 address and mask.
[dscp]	*DscpSpec* see 2.241.1.2.2.3	DSCP filter specification.
[dstPort]	*PortSpec* see 2.241.1.2.2.12	Destination port number filter specification.
[ecn] **	*EcnSpec* see 2.241.1.2.2.4	ECN filter specification. **Since: 4.17.0
[established]	*boolean*	True if filtered by established TCP connection.
[etherType] **	*integer*	Ethernet header's Ethernet type. **Since: 4.27.2
fragments	*boolean*	Match only fragmented packets.
gre	*GreSpec* see 2.241.1.2.2.5	GRE filter speciication.
gtp	*GtpSpec* see 2.241.1.2.2.6	GTP filter specification.
[icmp]	*IcmpSpec* see 2.241.1.2.2.7	ICMP specification.
[innerDest] **	*IpAddrAndMask* see 2.241.1.2.2.9	Inner destination IPv4 address and mask. **Since: 4.20.0
[innerDest6] **	*Ip6AddrAndMask* see 2.241.1.2.2.8	Inner destination IPv6 address and mask length. **Since: 4.20.0
[innerProtocol] **	*integer*	IP inner protocol number. **Since: 4.22.0
[innerSource] **	*IpAddrAndMask* see 2.241.1.2.2.9	Inner source IPv4 address and mask. **Since: 4.20.0
[innerSource6] **	*Ip6AddrAndMask* see 2.241.1.2.2.8	Inner source IPv6 address and mask length. **Since: 4.20.0
ipLen **	*IpLenSpec* see 2.241.1.2.2.10	IP packet length specification. **Since: 4.17.0
nexthopGroup **	*string*	Nexthop-group name. **Since: 4.17.0
nvgre	*NvgreSpec* see 2.241.1.2.2.11	NVGRE filter speciication.
protoMask **	*integer*	IP protocol mask. **Since: 4.27.2
protocol	*integer*	IP protocol number.
source	*IpAddrAndMask* see 2.241.1.2.2.9	Source IPv4 address and mask.
[srcPort]	*PortSpec* see 2.241.1.2.2.12	Source port number filter specification.

standard	*boolean*	True if this is an ACL in standard form.
[tcpFlags]	*integer*	TCP flags.
tracked	*boolean*	Connection tracked.
[ttl]	*TtlSpec* see 2.241.1.2.2.13	TTL specification.
userL4	*UserL4Spec* see 2.241.1.2.2.14	DPI based first word payload filter for unknown L4.
[vlan]	*VlanSpec* see 2.241.1.2.2.15	VLAN specification.
vxlan	*VxlanSpec* see 2.241.1.2.2.16	VXLAN filter specification.

2.241.1.2.2.3 DscpSpec

mask **	*integer*	DSCP mask. **Since: 4.23.2
match	*boolean*	True if filtered by DSCP value or name.
[name]	*string*	DSCP filter name.
value	*integer*	DSCP filter value.

2.241.1.2.2.4 EcnSpec

match	*boolean*	True if filtered by ECN value or name.
name	*enum: non-ect | ce | ect | dontCare | ect-ce*	ECN filter value.

2.241.1.2.2.5 GreSpec

protoMask	*integer*	GRE protocol mask.
protocol	*integer*	GRE protocol.

2.241.1.2.2.6 GtpSpec

[protocol]	*enum: gtp-c | gtp-prime | gtp-u*	GTP protocol type.
teid	*integer*	Tunnel Endpoint Id.
teidMask	*integer*	TEID Mask.
[version]	*integer*	GTP version.

2.241.1.2.2.7 IcmpSpec

code	*integer*	ICMP code.
type	*integer*	ICMP type.

2.241.1.2.2.8 Ip6AddrAndMask

[fullMask] **	*IPv6 address*	Ipv6 address full mask, used only when maskcannot be represented as a length. **Since: 4.20.5
ip	*IPv6 address*	Ipv6 address.
[mask]	*integer*	Ipv6 address mask len.

2.241.1.2.2.9 IpAddrAndMask

ip	*IPv4 address*	Ip address.
mask	*integer*	Ip address mask len.

2.241.1.2.2.10 IpLenSpec

ipLens	*list of integer*	IP packet lengths.
oper	*enum: lt | neq | gt | eq | range | any*	IP packet length range operator.

2.241.1.2.2.11 NvgreSpec

protoMask	*integer*	NVGRE protocol mask.
protocol	*integer*	NVGRE protocol.
tni	*integer*	Tenant Network Id.
tniMask	*integer*	TNI mask.

2.241.1.2.2.12 PortSpec

maxPorts	*integer*	Maximum number of ports.
oper	*enum: lt | neq | gt | eq | range | any*	Port number range operator.
ports	*list of integer*	Port numbers.

2.241.1.2.2.13 TtlSpec

oper	*enum: lt | neq | gt | eq | range | any*	TTL range operator.
value	*integer*	TTL value.

2.241.1.2.2.14 UserL4Spec

pattern	*integer*	Pattern to match into first word payload for IP packet.
patternMask	*integer*	Mask for pattern to match.

2.241.1.2.2.15 VlanSpec

id	*integer*	VLAN id.

innerId	*integer*	Inner VLAN id.
innerMask	*integer*	Inner VLAN mask.
mask	*integer*	VLAN mask.

2.241.1.2.2.16 VxlanSpec

vni	*integer*	Virtual Network Id.
vniMask	*integer*	VNI mask.
vxlanValid	*boolean*	True if valid VXLAN packet.

2.241.1.2.3 Ipv6Rule

[action]	*enum: permit | deny*	Action to take when the filter matches.
convertSymbols **	*boolean*	Auto-conversion from subnet and port numbers to words in show commands. **Since: 4.20.5
[copyToCpuDst] **	*enum: captive-portal | none*	Copy packet to CPU when this rule is matched. **Since: 4.25.1
[counterData]	*AclCounter* see 2.241.1.2.2.1	counter data for packets matching this rule.
[log]	*boolean*	Log when this rule is matched.
[mirrorSession] **	*string*	Target mirror session if this rule is matched. **Since: 4.23.0
[payload]	*PayloadSpec* see 2.52.1	DPI based payload filter specification.
[ruleFilter]	*Ipv6Filter* see 2.241.1.2.3.1	The filter criteria to match this rule. Unset for remarks. When an optional filter attribute is not present in the output, this implies that there was no specification configured for that attribute, and any valid value is permitted.
sequenceNumber	*integer*	Numeric sequence number.
text	*string*	A textual representation of the rule, sans sequence number. If this rule is a remark, this field will contain the remark text.

2.241.1.2.3.1 Ipv6Filter

destination	*Ip6AddrAndMask* see 2.241.1.2.2.8	Destination IPv6 address and mask.
[dscp]	*DscpSpec* see 2.241.1.2.2.3	DSCP filter specification.
[dstPort]	*PortSpec* see 2.241.1.2.2.12	Destination port number filter specification.
[ecn] **	*EcnSpec* see 2.241.1.2.2.4	ECN filter specification. **Since: 4.17.0
[established]	*boolean*	True if filtered by established TCP connection.

[etherType] **	*integer*	Ethernet header's Ethernet type. **Since: 4.27.2
[flowLabel] **	*FlowLabelSpec* see 2.241.1.2.3.2	IPv6 flow-label filter specification. **Since: 4.27.1
gre	*GreSpec* see 2.241.1.2.2.5	GRE filter speciication.
gtp	*GtpSpec* see 2.241.1.2.2.6	GTP filter specification.
[icmp]	*IcmpSpec* see 2.241.1.2.2.7	ICMP specification.
[innerDest] **	*IpAddrAndMask* see 2.241.1.2.2.9	Inner destination IPv4 address and mask. **Since: 4.20.0
[innerDest6] **	*Ip6AddrAndMask* see 2.241.1.2.2.8	Inner destination IPv6 address and mask length. **Since: 4.20.0
[innerProtocol] **	*integer*	IP inner protocol number. **Since: 4.22.0
[innerSource] **	*IpAddrAndMask* see 2.241.1.2.2.9	Inner source IPv4 address and mask. **Since: 4.20.0
[innerSource6] **	*Ip6AddrAndMask* see 2.241.1.2.2.8	Inner source IPv6 address and mask length. **Since: 4.20.0
ipLen **	*IpLenSpec* see 2.241.1.2.2.10	IP packet length specification. **Since: 4.17.0
nexthopGroup **	*string*	Nexthop-group name. **Since: 4.17.0
nvgre	*NvgreSpec* see 2.241.1.2.2.11	NVGRE filter speciication.
protoMask **	*integer*	IP protocol mask. **Since: 4.27.2
protocol	*integer*	IP protocol number.
source	*Ip6AddrAndMask* see 2.241.1.2.2.8	Source IPv6 address and mask.
[srcPort]	*PortSpec* see 2.241.1.2.2.12	Source port number filter specification.
standard	*boolean*	True if this is an ACL in standard form.
[tcpFlags]	*integer*	TCP flags.
tracked	*boolean*	Connection tracked.
[ttl]	*TtlSpec* see 2.241.1.2.2.13	TTL specification.
userL4	*UserL4Spec* see 2.241.1.2.2.14	DPI based first word payload filter for unknown L4.
[vlan]	*VlanSpec* see 2.241.1.2.2.15	VLAN specification.
vxlan	*VxlanSpec* see 2.241.1.2.2.16	VXLAN filter specification.

2.241.1.2.3.2 FlowLabelSpec

[mask] **	*integer*	IPv6 flow label mask. **Since: 4.27.2
oper	*enum: lt | neq | gt | eq | range | any | mask*	IPv6 flow-label range operator.

value	*integer*	IPv6 flow-label value.

2.241.1.2.4 MacRule

[action]	*enum: permit | deny*	Action to take when the filter matches.
convertSymbols **	*boolean*	Auto-conversion from subnet and port numbers to words in show commands. **Since: 4.20.5
[copyToCpuDst] **	*enum: captive-portal | none*	Copy packet to CPU when this rule is matched. **Since: 4.25.1
[counterData]	*AclCounter* see 2.241.1.2.2.1	counter data for packets matching this rule.
[log]	*boolean*	Log when this rule is matched.
[mirrorSession] **	*string*	Target mirror session if this rule is matched. **Since: 4.23.0
[payload]	*PayloadSpec* see 2.52.1	DPI based payload filter specification.
[ruleFilter]	*MacFilter* see 2.241.1.2.4.1	The filter criteria to match this rule. Unset for remarks. When an optional filter attribute is not present in the output, this implies that there was no specification configured for that attribute, and any valid value is permitted.
sequenceNumber	*integer*	Numeric sequence number.
text	*string*	A textual representation of the rule, sans sequence number. If this rule is a remark, this field will contain the remark text.

2.241.1.2.4.1 MacFilter

destinationAddr	*MAC address*	Destination address.
destinationMask	*MAC address*	Destination mask.
mpls	*MplsSpec* see 2.241.1.2.4.4	Two-label syntax MPLS filter specification.
[mplsFilter] **	*MplsFilter* see 2.241.1.2.4.2	MPLS filter specification. **Since: 4.20.0
protocol	*integer*	Ethernet protocol.
sourceAddr	*MAC address*	Source address.
sourceMask	*MAC address*	Source mask.
[vlan]	*VlanSpec* see 2.241.1.2.2.15	VLAN specification.

2.241.1.2.4.2 MplsFilter

[bos]	*integer*	MPLS BOS (Bottom Of Stack) bit of the largest specified label.
labels	*table(numeric string) = MplsLabelFilter* see 2.241.1.2.4.3	MPLS labels filter.

2.241.1.2.4.3 MplsLabelFilter

mask	*integer*	MPLS label mask.
value	*integer*	MPLS label value.

2.241.1.2.4.4 MplsSpec

innerLabel	*integer*	MPLS inner label id.
innerLabelMask	*integer*	MPLS inner label mask.
label	*integer*	MPLS label id.
labelMask	*integer*	MPLS label mask.

2.241.1.2.5 MplsRule

[counterData]	*ClassMapCounter* see 2.241.1.1	Number of packets that matched this class-map.

2.242 show class-map type tapagg [...]

see 2.241

2.243 show cli commands [({ hidden | functions | (truncate LENGTH) | verbatim })]

clis	*list of string*	CLI commands in the current mode.

2.244 show cli commands all-modes [({ hidden | functions | (truncate LENGTH) | verbatim })]

allClis	*table(string) = Clis* see 2.243	CLI commands in all modes, keyed by mode name.

2.245 show cli pid

pid	*integer*	PID of the current CLI.

2.246 show clock [...]

clockSource	*ClockSourceModel* see 2.246.1	Clock source.
localTime	*LocalTimeModel* see 2.246.2	Local time components.
timezone	*string*	Local timezone in use.
utcTime	*floating point number*	The current timestamp (UTC).

2.246.1 ClockSourceModel

local	*boolean*	Clock source is local.
[ntpServer]	*IP address*	NTP server address.
[ptpGrandmaster] **	*string*	PTP grandmaster clock identity. **Since: 4.26.2

2.246.2 LocalTimeModel

dayOfMonth	*integer*	Day of month (1 - 31).
dayOfWeek	*integer*	Day of week (0 - 6, Monday is 0).
dayOfYear	*integer*	Day of year (1 - 366).
daylightSavingsAdjust	*integer*	Daylight savings time (-1 - 1, -1 unknown).
hour	*integer*	Hour (0 - 23).
min	*integer*	Minute (0 - 59).
month	*integer*	Month (1 - 12).
sec	*integer*	Second ((0 - 61, 60 for leap secs, 61 for double leap secs).
year	*integer*	Year (for example, 2019).

2.247 show command-tag [...]

commandTags	*table(string) = CommandTagStatus* see 2.247.1	A mapping of command-tag to its status.

2.247.1 CommandTagStatus

operState	*enum: disabled | removed | enabled*	Command tag operational state.

2.248 show configuration checkpoints

checkpoints	*table(string) = Checkpoint* see 2.248.1	A mapping between the checkpoint name and info.
maxCheckpoints	*integer*	Maximum number of checkpoints.

2.248.1 Checkpoint

fileDate	*floating point number*	The utc timestamp of when the checkpoitn file created.
fileUser	*string*	The user who created the checkpoint file.

2.249 show configuration consistency nat [...]

tables	*table(string) = ConfigConsistencyTableModel* see 2.249.1	Config check result tables, keyed by table names.

2.249.1 ConfigConsistencyTableModel

headings	*list of string*	Table header fot this table.
items	*table(string) = ConfigConsistencyItemModel* see 2.249.1.1	Config consistency items, keyed by item names.
name	*string*	Name of table.
status	*enum: warn | ok | fail*	Table consistency level.

2.249.1.1 ConfigConsistencyItemModel

description	*string*	Description about the consistency item.
name	*string*	Name of item.
status	*enum: warn | ok | fail*	Item consistency level.

2.250 show configuration lock

[userInfo]	*ConfigureLockModelUser* see 2.250.1	Information for the user who holds the Configuration Lock.

2.250.1 ConfigureLockModelUser

[lockAcquireReason]	*string*	Reason why the lock was acquired.
lockAcquireTime	*floating point number*	UTC timestamp of when the lock was acquired.
[transactionName] **	*string*	Transaction name if a transaction lock. **Since: 4.28.1
[userLocation]	*string*	Source IP address or hostname for VTY sessions and absent for serial connections.
userTty	*string*	TTY for the user who acquired the lock.
username	*string*	The username of who acquired the lock.

2.251 show configuration lock history

history	*list of ConfigureLockModelHistoryInfo* see 2.251.1	Information for the previous users who held the Configuration Lock.

2.251.1 ConfigureLockModelHistoryInfo

lockAcquireTime	*floating point number*	UTC timestamp of when the lock was acquired.
[lockReleaseReason]	*string*	Reason why the lock was released.
[lockReleaseTime]	*floating point number*	UTC timestamp of when the lock was released.
[transactionName] **	*string*	Transaction name if a transaction lock. **Since: 4.28.1
[userLocation]	*string*	Source IP address or hostname for VTY sessions and absent for serial connections.
userTty	*string*	TTY for the user who held the lock.
username	*string*	The username of who held the lock.

2.252 show configuration sessions [...]

maxOpenSessions	*integer*	Maximum number of open sessions.
maxSavedSessions	*integer*	Maximum number of saved sessions.
mergeOnCommit **	*boolean*	New sessions will perform a commit merge. **Since: 4.29.1
sessions	*table(string) = Session* see 2.252.1	A mapping between the Session name and information.

2.252.1 Session

[commitBy] **	*floating point number*	Timestamp by when the session needs to be committed. **Since: 4.18.0
[commitUser] **	*string*	The user that committed this session. **Since: 4.26.2
[completedTime] **	*floating point number*	Timestamp of when the session completed. **Since: 4.26.2
description	*string*	Description of the session.
instances	*table(numeric string) = Instance* see 2.252.1.1	Instances where the session is open, indexed by process ID.
state	*enum: aborted | completed | pending | pendingCommitTimer*	State of the session.

2.252.1.1 Instance

currentTerminal	*boolean*	The session corresponds to the current terminal.
terminal	*string*	Terminal where the session is open.

user	*string*	User that opened the session.

2.253 show configuration sessions memory

sessions	*table(string) = SessionMemory* see 2.253.1	A mapping between the Session name and information.
totalMemory	*integer*	Memory in bytes used by all sessions.

2.253.1 SessionMemory

memory	*integer*	Memory in bytes used by the session.

2.254 show container-manager backup

backup	*table(string) = ContainerMgrBackupFiles* see 2.254.1	Container manager backup keyed by persistent directory.

2.254.1 ContainerMgrBackupFiles

backupFiles	*list of string*	Backup files.

2.255 show container-manager containers [...]

containerMgrContainers	*table(string) = ContainerMgrContainer* see 2.255.1	Container Manager containers keyed by containerName.

2.255.1 ContainerMgrContainer

command	*string*	Command for the container.
containerId	*string*	Container id.
imageId	*string*	Image id in sha256 format.
imageName	*string*	Image name with tag.
onBoot	*boolean*	Container will persist across reboots.
ports	*list of ContainerMgrPort* see 2.255.1.1	List of all the ports for this container.
state	*enum: dead | removing | exited | running | created | paused | restarting*	State of the container.
timeOfCreation	*integer*	Time of container creation.

2.255.1.1 ContainerMgrPort

ip	*IP address*	IP address of the port.
portType	*enum: tcp | udp*	Port type.
privatePort	*integer*	Private port number.
publicPort	*integer*	Public port number.

2.256 show container-manager images [...]

containerMgrImages	*table(string) = ContainerMgrImage* see 2.256.1	Container Manager images keyed by imageName:tag.

2.256.1 ContainerMgrImage

imageId	*string*	Image id in sha256 format.
imageSize	*integer*	Size of the image.
timeOfCreation	*integer*	Time of image creation.

2.257 show container-manager info

backingFilesystem	*string*	backing Filesystem.
cgroupDriver	*string*	cgroup Driver.
containerMgrEngineId	*string*	Container-manager engine ID.
containerMgrRootDir	*string*	Container-manager root dir.
containerNum	*integer*	Total number of Containers.
cpuNum	*integer*	Total number of CPUs.
hostName	*string*	hostname.
imagesNum	*integer*	Total number of images.
loggingDriver	*string*	logging Driver.
memory	*integer*	Total Memory.
networkType	*list of string*	Network types.
pausedContainerNum	*integer*	Total number of paused Containers.
runningContainerNum	*integer*	Total number of running Containers.
stoppedContainerNum	*integer*	Total number of stopped Containers.
storageDriver	*string*	storage Driver.
volumeType	*list of string*	Volume types.

2.258 show container-manager log CONTAINER_NAME

containerMgrLogs	*string*	Log of a container.

2.259 show container-manager registry [...]

containerMgrRegistries	*table(string) = ContainerMgrRegistry* see 2.259.1	Container Manager registry keyed by registryName.

2.259.1 ContainerMgrRegistry

insecure	*boolean*	Insecure registry.
serverName	*string*	Registry server name.
status	*enum: Success | Failed*	Registry status.
username	*string*	Username for registry.

2.260 show cpu counters queue

classes	*table(string) = MakoCpuCountersQueue-CoppClassCounts* see 2.260.1	Mapping between Copp class and total counts.

2.260.1 MakoCpuCountersQueueCoppClassCounts

packetsDropped	*integer*	Number of packets dropped belonging to this Copp class.
packetsTotal **	*integer*	Number of packets received (both admitted and dropped) belonging to this Copp class. **Since: 4.29.2

2.261 show cpu counters queue [...]

egressQueues	*EgressCpuCounters* see 2.261.1	CPU queue counters on the egress chip.
[ingressVoqs]	*IngressCpuCounters* see 2.261.2	CPU VOQ counters on the ingress source.

2.261.1 EgressCpuCounters

sources	*table(string) = EgressCpuPortCounters* see 2.261.1.1	Mapping between sources and CPU queue counters.

2.261.1.1 EgressCpuPortCounters

cpuPorts	*table(string) = EgressCpuDestinationTypeCounters* see 2.261.1.1.1	Mapping between CPU ports and queues.

2.261.1.1.1 EgressCpuDestinationTypeCounters

mcastQueues	*EgressCpuQueueCounters* see 2.261.1.1.1.1	Multicast CPU traffic class counters.
[ucastQueues]	*EgressCpuQueueCounters* see 2.261.1.1.1.1	Unicast CPU traffic class counters.

2.261.1.1.1.1 EgressCpuQueueCounters

queues	*table(string) = Counters* see 2.261.1.1.1.2	Mapping between CPU queue and counter data.

2.261.1.1.1.2 Counters

[droppedBitsRate] **	*floating point number*	Bits dropped per second. **Since: 4.22.0
[droppedBytes]	*integer*	Octets (bytes) dropped.
[droppedPackets]	*integer*	Packets dropped.
[droppedPacketsRate] **	*floating point number*	Packets dropped per second. **Since: 4.22.0
[enqueuedBitsRate] **	*floating point number*	Bits enqueued per second. **Since: 4.22.0
[enqueuedBytes]	*integer*	Octets (bytes) enqueued.
[enqueuedPackets]	*integer*	Packets enqueued.
[enqueuedPacketsRate] **	*floating point number*	Packets enqueued per second. **Since: 4.22.0

2.261.2 IngressCpuCounters

sources	*table(string) = IngressCpuClassCounters* see 2.261.2.1	Mapping between source and CPU class.

2.261.2.1 IngressCpuClassCounters

`cpuClasses`	*table(string) = IngressCpuPortCounters* see 2.261.2.1.1	Mapping between CPU classes and ingress port.

2.261.2.1.1 IngressCpuPortCounters

`ports`	*table(string) = Counters* see 2.261.1.1.1.2	Mapping between ingress port and corresponding counter data.

2.262 show cvx [...]

revision 2

clusterMode	*boolean*	CVX server in cluster mode.
[clusterStatus]	*ClusterStatus* see 2.262.1	Cluster status.
connectionPreserved **	*boolean*	Client connection state preserving is enabled. **Since: 4.22.0
controllerUUID	*string*	Controller UUID.
enabled	*boolean*	Status of the CVX server.
heartbeatInterval	*floating point number*	Heartbeat interval in seconds.
heartbeatTimeout	*floating point number*	Heartbeat timeout in seconds.
[sslProfileStatus]	*SslProfileStatus* see 2.262.3	SSL profile status.

2.262.1 ClusterStatus

Could also be an instance of:
ClusterStatusDebug 2.262.2

clusterName	*string*	Name.
[currLeader]	*string*	Leader.
lastLeaderChangeTs	*floating point number*	Cluster's last leadership change timestamp.
peerStatus	*table(string) = PeerStatus* see 2.262.1.1	Cluster's peer status.
peerTimeout	*floating point number*	Heartbeat timeout in seconds for intra cluster communication.
role	*enum: Standby | Disconnected | Master*	Controller Role.

2.262.1.1 PeerStatus

Could also be an instance of:
PeerStatusDebug 2.262.1.2

peerIp	*IPv4 address*	Peer IP address.
peerName	*string*	Peer name.
[peerSystemId]	*string*	Peer Id.
[registrationErrorStatus]	*string*	Peer registration error description.
registrationState	*string*	Peer registration state.
[serviceVersionStatus]	*string*	Peer service version compatibility.

2.262.1.2 PeerStatusDebug

peerIp	*IPv4 address*	Peer IP address.
peerName	*string*	Peer name.
[peerSystemId]	*string*	Peer Id.
[registrationErrorStatus]	*string*	Peer registration error description.
registrationState	*string*	Peer registration state.
[serviceVersionStatus]	*string*	Peer service version compatibility.

sshKeyCopyStatus	*string*	Ssh key copy status.

2.262.2 ClusterStatusDebug

clusterName	*string*	Name.
[currLeader]	*string*	Leader.
currState	*string*	Election State.
currTerm	*string*	Cluster's current term.
lastLeaderChangeTerm	*string*	Cluster's last leadership change term.
lastLeaderChangeTs	*floating point number*	Cluster's last leadership change timestamp.
peerStatus	*table(string) = PeerStatus* see 2.262.1.1	Cluster's peer status.
peerTimeout	*floating point number*	Heartbeat timeout in seconds for intra cluster communication.
role	*enum: Standby | Disconnected | Master*	Controller Role.

2.262.3 SslProfileStatus

[disableReason]	*enum: profileNotExist | dhparamsNotReady | trustedCertNotConfigured | certNotConfigured | profileInvalid | noError*	Reason for disabled SSL profile.
enabled	*boolean*	Whether SSL profile is enabled.
sslProfileName	*string*	SSL profile name.

2.263 show cvx connections [...]

revision 2

brief **	*boolean*	Show brief status only. **Since: 4.22.0
connections	*list of CvxConnection* see 2.263.1	Cvx connections.

2.263.1 CvxConnection

[connectionTime] **	*floating point number*	Connection establishment UTC timestamp. **Since: 4.18.0
heartbeatStatus	*HeartbeatStatus* see 2.263.1.1	Heartbeat status of the switch.
hostname	*string*	Switch hostname.
[ibConnectionSecured]	*boolean*	Whether in-band connection is SSL secured.
oobConnectionActive **	*boolean*	Connection state. **Since: 4.22.0
[oobConnectionSecured]	*boolean*	Whether out-of-band connection is SSL secured.

`oobState` **	*enum: connecting | negotiating | disconnecting | established*	Out-of-band state. **Since: 4.15.4
`switchId`	*MAC address*	Switch Id.

2.263.1.1 HeartbeatStatus

`lastHeartbeatReceived`	*floating point number*	Last heartbeat.
`lastHeartbeatSent`	*floating point number*	Last heartbeat sent.

2.264 show cvx connections client (SYSTEMID | HOSTNAME)

revision 2

see 2.263.1

2.265 show cvx connections client (SYSTEMID | HOSTNAME) brief

revision 2

see 2.263

2.266 show cvx debug dirs [...]

`[dirChurnCount]`	*integer*	Dir churn count.
`[dirStatus]`	*table(string) = DirStatus* see 2.266.1	Mapping from dir name to status.
`[entityChurnCount]`	*integer*	Entity churn count.

2.266.1 DirStatus

`entityChurnCount`	*integer*	Entity churn count.
`[entityStatus]`	*table(string) = EntityStatus* see 2.266.1.1	Mapping from entity name to status.
`entryCount`	*integer*	Entry count.

2.266.1.1 EntityStatus

`stringBufChurnCount`	*integer*	Strung buffer churn count.
`stringBufCount`	*integer*	String buffer count.

2.267 show cvx debug pingpong

`addsPerSecond`	*integer*	Adds per second.
`addsPerSecondSnapshot`	*integer*	Last adds per second snapshot.

[delsPerSecond]	*integer*	Last deletes per second snapshot.
[dirChangeCount]	*integer*	Number of reactor fires on pingpong directories.
[downtimeCount]	*integer*	Number of downtime periods.
enabled	*boolean*	Ping test is enabled.
[longestDowntime]	*floating point number*	Longest downtime (seconds).
runTime	*floating point number*	Total run time (seconds).
[totalDataChangeCount]	*integer*	Number of reactor fires on entities in pingpong directories.
[totalDowntime]	*floating point number*	Total downtime (seconds).
totalPings	*integer*	Total pings.

2.268 show cvx file-replication

services	*table(string) = Status* see 2.268.1	A mapping from service name to sets of replicated files.

2.268.1 Status

requesters	*table(string) = RequesterStatus* see 2.268.1.1	A mapping from replicated file set names to the the replication status of the file set.

2.268.1.1 RequesterStatus

files	*table(string) = FileStatus* see 2.268.1.1.1	A mapping of file names to the replication status of each file.

2.268.1.1.1 FileStatus

[lastSyncTime]	*floating point number*	Last synchronized time.
status	*enum: unsynchronized | failed | synchronized*	Status of file synchronization.

2.269 show cvx file-replication NAME

see 2.268.1

2.270 show cvx mounts

connections	*list of CvxMounts* see 2.270.1	Cvx mounts for all connections.

2.270.1 CvxMounts

hostname **	*string*	Switch hostname. **Since: 4.23.2
mounts	*list of ServiceMountStates* see 2.270.1.1	CVX service mount states.

switchId	*MAC address*	Switch Id.

2.270.1.1 ServiceMountStates

mountStates	*list of GroupMountStates* see 2.270.1.1.1	States of groups of mounts.
service	*string*	Service.

2.270.1.1.1 GroupMountStates

group	*string*	Group.
pathStates	*list of MountState* see 2.270.1.1.1.1	Publish states of paths within a group.
state	*string*	Publish state of group.
systemWhitelist	*list of string*	System whitelist.
versionWhitelist	*list of integer*	Version whitelist.

2.270.1.1.1.1 MountState

path	*string*	Mount path.
state	*string*	Mount state.
type	*string*	Mount type.

2.271 show cvx mounts (HOSTNAME | SYSTEMID)

see 2.270.1

2.272 show cvx service

revision 2

services	*list of CvxService* see 2.272.1	Status of all services running on the controller.

2.272.1 CvxService

name	*string*	Service name.
status	*boolean*	Service status.
switches	*table(string) = ServiceStatus* see 2.272.1.1	Status of all the switches running the service.
versions	*list of integer*	Supported versions.

2.272.1.1 ServiceStatus

enabled	*boolean*	If the service is enabled.
negotiatedVersion	*integer*	Negotiated version.
versionCompatible	*boolean*	Version compatibility.
versionNegotiationComplete	*boolean*	Status of negotiation process.

2.273 show cvx service NAME

revision 2

see 2.272.1

2.274 show cvx service peer [...]

clusterPeers	*table(string) = CvxPeerServiceStatus* see 2.274.1	A mapping of peer name to its service status.

2.274.1 CvxPeerServiceStatus

isLeader	*boolean*	Peer is leader.
overallCompatibility	*enum: ok | partial | unknown | mismatch*	Overall service compatibility for peer.
services	*table(string) = CvxPeerServiceCompatibility* see 2.274.1.1	A mapping of service name to its compatibility information.

2.274.1.1 CvxPeerServiceCompatibility

compatibility	*enum: disabledBoth | versionMismatch | unknownPeer | unknownLocal | disabledLocal | disabledPeer | compatible*	Service compatibility status.
[details]	*string*	Service compatibility details.

2.275 show daemon DAEMON_NAME ((ip access-list) | (ipv6 access-list) | (access-list)) [...]

see 2.53

2.276 show daemon [...]

daemons	*table(string) = DaemonAgent* see 2.276.1	A mapping from the daemon process name to its configuration.

2.276.1 DaemonAgent

data	*table(string) = string*	A string/string map of status data published by the process.
enabled	*boolean*	Whether the daemon process is configured to run.
[isSdkAgent]	*boolean*	True if this process contains an executable that uses EOS SDK to run a third-party agent. This field will be omitted if the agent is not running.
option	*table(string) = string*	A string/string map of options configured for the process.
[pid] **	*integer*	The PID of the process, if the agent is running. **Since: 4.20.5
running	*boolean*	True if this process is actively running.
[starttime] **	*floating point number*	The start time of the process, if the agent is running. **Since: 4.21.0
[uptime] **	*floating point number*	The uptime of the process, if the agent is running. **Since: 4.21.0

2.277 show dcbx [...]

etsConfig **	*EtsTlvModel* see 2.277.2	ETS configuration TLV. **Since: 4.30.0
etsRecommendation **	*EtsTlvModel* see 2.277.2	ETS recommendation TLV. **Since: 4.30.0
interfaces	*table(Interface name) = InterfaceDescription* see 2.277.1	Map to interface class which holds all of the interface metadata.

2.277.1 InterfaceDescription

apcTlvMalformed	*integer*	Number of Application priority configuration TLVs that are malformed.
apcTlvReceived	*integer*	Number of Application priority configuration TLVs received.
apciEntries	*table(string) = ApplicationPriorityEntryData* see 2.277.1.1	Application priority configuration(apci) object. This object contains metadata for every apci entry. There is a one-to-one mapping between each entry protocol ID and the enabled priorities stored within each value.
dcbxEnabled	*boolean*	Interface exists in the DCBX configuration scheme.
ieeeTlvReceived	*boolean*	Ieee TLVs(Type-Lengh-Values) have been received.
lastLldpduTimestamp	*floating point number*	Unix time stamp taken whenever an LLDPDU is received.
macSecBypassCapability	*boolean*	Capable of bypassing MACsec.
pfcPortConfigEnabled	*boolean*	Port PFC configured.
pfcPortConfigPriorities	*integer*	Number of PFC configuration priorities.
pfcTlvMalformed	*integer*	Number of PFC(priority flow control) TlVs that are malformed.
pfcTlvReceived	*integer*	Number of PFC(priority flow control) TlVs received.
pfcWilling	*boolean*	PFC configuration.
pfcciPfcCapability	*integer*	PFC capabilities.
pfcciPfcEnable	*integer*	PFC enabled priorities.
rxEnabled	*boolean*	LLDP adminstatus is enabled for receive.
txEnabled	*boolean*	LLDP(Link Layer Discovery Protocol) adminstatus is enabled for transmit.

2.277.1.1 ApplicationPriorityEntryData

priority	*list of integer*	Corresponding protocol ID enabled priority.
protocolId	*list of integer*	Protocol ID that maps to the protocol's priority.

2.277.2 EtsTlvModel

trafficClasses	*list of TrafficClassModel* see 2.277.2.1	Traffic classes data.

2.277.2.1 TrafficClassModel

algorithm	*integer*	Algorithm identifier.
bandwidth	*integer*	Bandwidth value.
priority	*integer*	Traffic class priority.

2.278 show debugging counter { COUNTERS }

counters	*table(string) = integer*	A mapping between counter name and counter value.
timestamp	*floating point number*	Pulse Counter timestamp.

2.279 show debugging counters [...]

see 2.278

2.280 show debugging inotify (QUERY | all)

[count]	*integer*	Count of total inotify FDs currently allocated.
[limits]	*table(string) = integer*	A mapping of the inotify limit to its value.
[ps]	*table(string) = InotifyProc* see 2.280.2	A mapping of the /proc/PID/fd of each process using inotify to its process information.
[user]	*table(numeric string) = InotifyProc* see 2.280.2	A mapping of the number of inotify FDs in use by each user.
[watch] **	*InotifyPids* see 2.280.1	Inotify watches grouped by PID. **Since: 4.29.2

2.280.1 InotifyPids

pids	*table(numeric string) = InotifyPid* see 2.280.1.1	A mapping from a process ID to its inotify instances.

2.280.1.1 InotifyPid

[cmdline]	*string*	The /proc/PID/cmdline of the process.
fds	*table(numeric string) = InotifyFiles* see 2.280.1.1.1	A mapping from an inotify instance file descriptor to its watched files.

2.280.1.1.1 InotifyFiles

files	*list of InotifyFile* see 2.280.1.1.1.1	Filepaths and corresponding watch descriptors.

2.280.1.1.1.1 InotifyFile

filepath	*string*	The path of the file being watched.
watchDescriptor	*integer*	The watch descriptor associated with the watch.

2.280.2 InotifyProc

[cmdline]	*string*	The /proc/PID/cmdline of the process.
count	*integer*	The number of inotify FDs owned by the process.
[owner]	*string*	The owner of the process.

2.281 show debugging inotify file pid PID

files	*list of InotifyFile* see 2.280.1.1.1.1	Files in var/shmem being watched by the process using inotify.

2.282 show debugging shark clients all

2.283 show debugging smash clients all

[paths]	*table(string) = TableInfo* see 2.283.1	A mapping of the absolute path of the Smash table to information about this table.

2.283.1 TableInfo

headerInfo	*HeaderInfo* see 2.283.1.1	Header information of this Smash table.
[keyshadow]	*list of string*	All keyshadows of this Smash table.
[reader]	*list of string*	All readers of this Smash table.
[shadow]	*list of string*	All shadows of this Smash table.
[writer]	*string*	The writer of this Smash table.

2.283.1.1 HeaderInfo

[capacity]	*CapacityInfo* see 2.283.1.1.1	Current capacity information.
[occupiedSlots]	*integer*	Current number of occupied slots.

2.283.1.1.1 CapacityInfo

[buckets]	*integer*	Current buckets capacity.
[primaryNotifications]	*integer*	Current capacity of the primary conquer queue.
[secondaryNotifications]	*integer*	Current capacity of the secondary conquer queue.
[slots]	*integer*	Current slots capacity.

2.284 show deep-inspection summary

l2SkipValue	*integer*	deep-inspection payload skip value for L2.
l4SkipValue	*integer*	deep-inspection payload skip value for L4.

2.285 show dhcp server ((ipv4 leases) | (ipv6 leases) | leases) [...]

vrfs	*table(string) = DhcpServerShowLeasesModel* see 2.285.1	DHCP server lease information for all vrfs.

2.285.1 DhcpServerShowLeasesModel

ipv4ActiveLeases	*list of Ipv4Leases* see 2.285.1.1	Active IPv4 Leases.
ipv4LeaseError **	*boolean*	There was an error grabbing IPv4 leases. **Since: 4.29.2
ipv6ActiveLeases **	*list of Ipv6Leases* see 2.285.1.2	Active IPv6 Leases. **Since: 4.23.0
ipv6LeaseError **	*boolean*	There was an error grabbing IPv4 leases. **Since: 4.29.2

2.285.1.1 Ipv4Leases

endLeaseTime	*floating point number*	End time of lease.
ipAddress	*IPv4 address*	IP address.
lastTransaction	*floating point number*	Last transaction time with client.
macAddress	*MAC address*	MAC address of client.
[option82] **	*Option82* see 2.285.1.1.1	Information Option. **Since: 4.28.2

2.285.1.1.1 Option82

[circuitIds]	*list of string*	Circuit ID in hex.
[remoteIds]	*list of string*	Remote ID in hex.
[unknownSubOpts]	*list of Option82SubOption* see 2.285.1.1.1.1	Unknown sub-options.

2.285.1.1.1.1 Option82SubOption

typeCode	*integer*	Unknown sub-option type code.
value	*string*	Unknown sub-option value in hex.

2.285.1.2 Ipv6Leases

endLeaseTime	*floating point number*	End time of lease.
ipAddress	*IPv6 address*	Ipv6 Address.
lastTransaction	*floating point number*	Last transaction time with client.
macAddress	*MAC address*	MAC address of client.

2.286 show dhcp server [...]

vrfs	*table(string) = DhcpServerShowModel* see 2.286.1	DHCP server information for all vrfs.

2.286.1 DhcpServerShowModel

[addressFamily] **	*string*	IP version of the server called. **Since: 4.23.0
ipv4Server	*DhcpServerIpv4ShowModel* see 2.286.1.1	IPv4 DHCP server information.
ipv6Server **	*DhcpServerIpv6ShowModel* see 2.286.1.2	IPv6 DHCP server information. **Since: 4.23.0

2.286.1.1 DhcpServerIpv4ShowModel

[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
[debugLog] **	*boolean*	Debug log is enabled. **Since: 4.23.0
ipv4ActiveLeases	*integer*	Number of active leases.
[ipv4DisabledMessage]	*string*	Why IPv4 DHCP server is disabled if configured.
ipv4DnsServers	*list of IPv4 address*	DNS servers.
ipv4DomainName **	*string*	DNS domain name. **Since: 4.23.0
ipv4Interfaces **	*table(Interface name) = DhcpServerInterfaceStatusShowModel* see 2.286.1.1.2	A mapping of IPv4 DHCP interfaces to their status. **Since: 4.22.1
ipv4LeaseDuration	*floating point number*	Duration of leases (seconds).
ipv4ServerActive	*boolean*	IPv4 DHCP server is active.
ipv4Subnets	*table(IPv4 or IPv6 address prefix) = DhcpServerIpv4SubnetModel* see 2.286.1.1.3	Dictionary of IPv4 subnets configured.
[ipv4TftpBootFile] **	*string*	TFTP boot file. **Since: 4.23.1
[ipv4TftpServerOption150] **	*list of IPv4 address*	TFTP Server option 150. **Since: 4.23.2
[ipv4TftpServerOption66] **	*string*	TFTP Server option 66. **Since: 4.23.1
[ipv4VendorOptions] **	*table(string) = DhcpServerIpv4VendorOptionModel* see 2.286.1.1.4	Dictionary of IPv4 vendor-options configured, mapping vendor-option to sub-options. **Since: 4.24.1

2.286.1.1.1 DhcpServerClientClassModel

inactiveReason	*string*	Inactive Reason.

2.286.1.1.2 DhcpServerInterfaceStatusShowModel

active	*boolean*	This interface is active.
[detail]	*string*	Details of interface activity.

2.286.1.1.3 DhcpServerIpv4SubnetModel

activeLeases	*integer*	Number of active leases for this subnet.
[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
defaultGateway	*IPv4 address*	Default gateway address.
[disabledMessage]	*string*	Why IPv4 subnet is disabled if configured.
dnsServers	*list of IPv4 address*	DNS servers.
[duplicateReservedIp] **	*IPv4 address*	Duplicate reserved IPv4 address. **Since: 4.25.2
[invalidRange] **	*DhcpServerIpv4Range* see 2.286.1.1.3.1	Invalid ranges. **Since: 4.25.2
[leaseDuration] **	*floating point number*	Duration of leases (seconds). **Since: 4.23.1
name **	*string*	Subnet name. **Since: 4.23.0
[overlappingRange] **	*DhcpServerIpv4Range* see 2.286.1.1.3.1	Overlapping ranges. **Since: 4.25.2
[overlappingSubnets] **	*list of IP address prefix*	Overlapping subnets. **Since: 4.25.0
ranges	*list of DhcpServerIpv4Range* see 2.286.1.1.3.1	Ranges configured.
[reservationsMacAddress] **	*table(MAC address) = DhcpServerIpv4ReservationsMacAddressModel* see 2.286.1.1.3.2	Dictionary of MAC addresses to reserved IPv4 addresses. **Since: 4.24.1
subnet	*IP address prefix*	Subnet.
[tftpBootFile] **	*string*	TFTP boot file. **Since: 4.23.1
[tftpServerOption150] **	*list of IPv4 address*	TFTP server option 150. **Since: 4.23.2
[tftpServerOption66] **	*string*	TFTP server option 66. **Since: 4.23.1
[unknownSubnet] **	*boolean*	Subnet is unknown. **Since: 4.25.2

2.286.1.1.3.1 DhcpServerIpv4Range

[assignedLeases] **	*integer*	Number of assigned leases in range. **Since: 4.28.2

[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
endRangeAddress	*IPv4 address*	End address.
startRangeAddress	*IPv4 address*	Start address.
[totalLeases] **	*integer*	Number of total leases in range. **Since: 4.28.2

2.286.1.1.3.2 DhcpServerIpv4ReservationsMacAddressModel

[hostname] **	*string*	Reserved hostname. **Since: 4.29.0
[ipv4Address]	*IPv4 address*	Reserved IPv4 address.
macAddress	*MAC address*	MAC address of client.

2.286.1.1.4 DhcpServerIpv4VendorOptionModel

subOptions	*list of DhcpServerSubOptionModel* see 2.286.1.1.4.1	Sub-options configured.

2.286.1.1.4.1 DhcpServerSubOptionModel

optionCode	*integer*	Sub-option code.
[optionDataIpAddresses]	*list of IPv4 address*	IPv4 addresses for sub-option type 'ipv4-address'.
[optionDataString]	*string*	String for sub-option type 'string'.
optionType	*enum: string | ipAddress*	Sub-option type.

2.286.1.2 DhcpServerIpv6ShowModel

[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
[debugLog]	*boolean*	Debug log is enabled.
ipv6ActiveLeases	*integer*	Number of active leases.
[ipv6DisabledMessage]	*string*	Why IPv6 DHCP server is disabled if configured.
ipv6DnsServers	*list of IPv6 address*	DNS servers.
ipv6DomainName	*string*	DNS domain name.
ipv6Interfaces	*table(Interface name) = DhcpServerInterfaceStatusShowModel* see 2.286.1.1.2	A mapping of IPv4 DHCP interfaces to their status.
ipv6LeaseDuration	*floating point number*	Duration of leases (seconds).
ipv6ServerActive	*boolean*	IPv6 DHCP server is active.
ipv6Subnets	*table(IPv4 or IPv6 address prefix) = DhcpServerIpv6SubnetModel* see 2.286.1.2.1	Dictionary of IPv6 subnets configured.

2.286.1.2.1 DhcpServerIpv6SubnetModel

activeLeases	*integer*	Number of active leases for this subnet.
[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
directActive	*boolean*	IPv6 server responds to direct requests.
[directActiveDetail]	*string*	Details of direct status.
[disabledMessage]	*string*	Why IPv6 subnet is disabled if configured.
dnsServers	*list of IPv6 address*	DNS servers.
[duplicateReservedIp] **	*IPv6 address*	Duplicate reserved IPv6 address. **Since: 4.27.2
[invalidRange] **	*DhcpServerIpv6Range* see 2.286.1.2.1.1	Invalid ranges. **Since: 4.25.2
[leaseDuration] **	*floating point number*	Duration of leases (seconds). **Since: 4.23.1
name	*string*	Subnet name.
[overlappingRange] **	*DhcpServerIpv6Range* see 2.286.1.2.1.1	Overlapping ranges. **Since: 4.25.2
[overlappingSubnets] **	*list of IP address prefix*	Overlapping subnets. **Since: 4.25.0
ranges	*list of DhcpServerIpv6Range* see 2.286.1.2.1.1	Ranges configured.
relayActive	*boolean*	IPv6 server responds to relay requests.
[relayActiveDetail]	*string*	Details of relay status.
[reservationsMacAddress] **	*table(MAC address) = DhcpServerIpv6ReservationsMacAddressModel* see 2.286.1.2.1.2	Dictionary of MAC addresses to reserved IPv6 addresses. **Since: 4.27.2
subnet	*IP address prefix*	Subnet.
[unknownSubnet] **	*boolean*	Subnet is unknown. **Since: 4.25.2

2.286.1.2.1.1 DhcpServerIpv6Range

[assignedLeases] **	*integer*	Number of assigned leases in range. **Since: 4.28.2
[clientClasses] **	*table(string) = DhcpServerClientClassModel* see 2.286.1.1.1	Dictionary of Client Class names to inactive reasons. **Since: 4.28.0
endRangeAddress	*IPv6 address*	End address.
startRangeAddress	*IPv6 address*	Start address.
[totalLeases] **	*integer*	Number of total leases in range. **Since: 4.28.2

2.286.1.2.1.2 DhcpServerIpv6ReservationsMacAddressModel

[hostname] **	*string*	Reserved hostname. **Since: 4.29.0
[ipv6Address]	*IPv6 address*	Reserved IPv6 address.

macAddress	*MAC address*	MAC address of client.

2.287 show direct-table [...]

configuredEntryCount	*integer*	Total configured entry count.
configuredTableCount	*integer*	Total configured table count.
disabled	*boolean*	Direct Table is disabled.
errorCount	*integer*	Count of programming errors.
pendingCount	*integer*	Count of pending entries.
programmedCount	*integer*	Count of entries in hardware.

2.288 show direct-table errors

tables	*table(string) = TableState* see 2.288.1	Mapping of a table to its entries.

2.288.1 TableState

tableEntries	*table(string) = EntryState* see 2.288.1.1	Mapping of an entry to its state.

2.288.1.1 EntryState

state	*enum: failedOther | inHardware | pending | failedNoResources | failedNoAction | failedAlreadyExists*	State of the entry.

2.289 show direct-table state

see 2.288

2.290 show direct-table table TABLE [...]

actionParameters	*table(string) = ActionParameters* see 2.290.1	A mapping of action to its parameters.
byteCounters	*boolean*	This table supports byte counters.
configured	*boolean*	The table is configured.
configuredEntryCount	*integer*	Total configured entry count.
errorCount	*integer*	Count of programming errors.
matchType	*enum: ternary | exact*	Table match type.
matches	*table(string) = string*	A mapping of match field to its description.
name	*string*	Name of this table.
packetCounters	*boolean*	This table supports packet counters.
pendingCount	*integer*	Count of pending entries.

programmedCount	*integer*	Count of entries in hardware.
size	*integer*	Maximum number of entries.

2.290.1 ActionParameters

parameters	*list of string*	Name of parameters.

2.291 show direct-table table TABLE counters

byteCounters	*boolean*	This table supports byte counters.
entries	*table(string) = EntryCounters* see 2.291.1	A mapping of entry to its counter values.
name	*string*	Name of this table.
packetCounters	*boolean*	This table supports packet counters.

2.291.1 EntryCounters

byteCount	*integer*	Total sum of bytes from packets that have hit this entry.
packetCount	*integer*	Number of packets that have hit this entry.

2.292 show direct-table table TABLE entries

entries	*table(string) = EntryDetails* see 2.292.1	Mapping of an entry name to its details.
fieldTypes	*table(string) = TableFieldType* see 2.292.3	Mapping of name to type for match fields in this table.
isTernary	*boolean*	This table uses ternary matches.
name	*string*	Name of this table.
parameterTypes	*table(string) = ParameterType* see 2.292.2	Mapping of name to type for action parameters in this table.

2.292.1 EntryDetails

[actionName]	*string*	The action name of this entry.
matchValues	*table(string) = GenericMatchAndMask* see 2.292.1.1	Mapping of match name to its value for this entry.
parameterValues	*table(string) = integer*	Mapping of parameter name to its value for this entry.
[priority]	*integer*	The priority of this entry.

2.292.1.1 GenericMatchAndMask

[integerValue]	*MatchValueAndMask* see 2.292.1.1.2	Integer match value.
[interfaceValue]	*interface name*	Interface match value.
[ipv4Prefix]	*IpAddrAndMask* see 2.241.1.2.2.9	IPv4 address prefix match value.
[ipv6Prefix]	*Ip6AddrAndMask* see 2.241.1.2.2.8	IPv6 address prefix match value.
[macAddress]	*MacAddressValueAndMask* see 2.292.1.1.1	MAC address match value.

2.292.1.1.1 MacAddressValueAndMask

[mask]	*MAC address*	MAC address mask.
value	*MAC address*	MAC address value.

2.292.1.1.2 MatchValueAndMask

[mask]	*integer*	Mask of this value.
value	*integer*	A match value.

2.292.2 ParameterType

parameterType	*integer*	Parameter value type.

2.292.3 TableFieldType

matchType	*enum: lpm | ternary | exact*	The match type of this field.
valueType	*enum: ipv6Prefix | integer | ipv4Prefix | interface | macAddress*	The type of value this match field has.

2.293 show direct-table table TABLE errors

see 2.288

2.294 show direct-table table TABLE state

see 2.288

2.295 show directflow

configuredFlows **	*integer*	Total configured flows. **Since: 4.26.0
enabled	*boolean*	DirectFlow enabled.
pendingFlows **	*integer*	Pending flows. **Since: 4.26.0
rejectedFlows **	*integer*	Rejected flows. **Since: 4.26.0
totalFlows **	*integer*	Programmed flows. **Since: 4.20.5
totalPacketCount	*integer*	Matched packets.

2.296 show directflow counters [...]

revision 2

enabled	*boolean*	DirectFlow enabled.
sources	*table(string) = Flows* see 2.296.1	Mapping between Dynamic flow source and Flows.
totalCounters	*boolean*	Total Counters to be printed.
totalPacketCount	*integer*	Total Matched packets.

2.296.1 Flows

flows	*list of Flow* see 2.296.1.1	Flows.

2.296.1.1 Flow

[action]	*Action* see 2.296.1.1.1	Action set values.
[bridgeMacAddr]	*string*	Bridge MAC Address.
[cookie]	*integer*	flow cookie.
[countersAvailable] **	*boolean*	Hardware counters available. **Since: 4.20.5
countersPerSlice **	*table(string) = Counter* see 2.296.1.1.2	Dictionary of counters keyed with slice name. **Since: 4.18.0
[hardTimeout]	*integer*	Hard timeout.
[idleTimeout]	*integer*	Idle timeout.
[installation] **	*boolean*	Install flow after resource allocation. **Since: 4.24.2
[match]	*Match* see 2.296.1.1.3	Match values.
[matchBytes]	*integer*	Number of bytes that matched flow.
[matchPackets]	*integer*	Number of packets that matched flow.
name	*string*	Flow name.
[persistent]	*boolean*	Flow persistence.
priority	*integer*	Flow priority.
[priorityGroupType] **	*enum: flood | vlan | ipv6Fib | ethdstHairpin | default | portacl | high | vlanacl | ipv4Fib | low | ethdst | vip | ethsrc*	Flow priority Group. **Since: 4.18.0
statusPerSlice **	*table(string) = Status* see 2.296.1.1.4	Dictionary of status's keyed with slice name. **Since: 4.18.0
[tableId] **	*integer*	Table Id. **Since: 4.25.0
[tableName] **	*string*	Flow name. **Since: 4.25.0
[tableType] **	*string*	Flow table type. **Since: 4.18.0

2.296.1.1.1 Action

[addVlanInner] **	*integer*	Add inner Vlan ID. **Since: 4.22.1
[addVlanInnerPCP] **	*integer*	Add inner Vlan PCP. **Since: 4.22.1
[addVlanOuter] **	*integer*	Add outer Vlan ID. **Since: 4.23.1
[addVlanOuterPCP] **	*integer*	Add outer Vlan PCP. **Since: 4.23.1
[debugCapture] **	*boolean*	Debug Capture. **Since: 4.18.0
[dropTag] **	*boolean*	Drop Outer Tag. **Since: 4.20.0
[egrMirrorInterfaces]	*list of Interface name*	Copy egress to mirror destination interfaces.
[goto] **	*boolean*	Goto Action. **Since: 4.23.0
[group]	*integer*	Forward to group.
[icmpCode] **	*integer*	Set ICMP code. **Since: 4.23.1
[icmpType] **	*integer*	Set ICMP type. **Since: 4.23.1
[ingrMirrorInterfaces]	*list of Interface name*	Copy ingress to mirror destination interfaces.
[ipDst] **	*IpAddr* see 2.296.1.1.1.4	Set destination IP address. **Since: 4.25.1
[ipPortDst]	*integer*	Set TCP/UDP destination port.
[ipPortSrc]	*integer*	Set TCP/UDP source port.
[ipSrc] **	*IpAddr* see 2.296.1.1.1.4	Set source IP address. **Since: 4.25.1
[ipTos]	*integer*	Set Ipv4 Tos (as received).
[localIntf] **	*string*	Local intf. **Since: 4.20.5
[loopback]	*boolean*	Output to ingress interface.
[macDst]	*MacAddr* see 2.296.1.1.1.3	Set destination Mac address.
[macDstBr]	*MacAddr* see 2.296.1.1.1.3	Set destination Mac address to router-address.
[macSrc]	*MacAddr* see 2.296.1.1.1.3	Set source Mac address.
[macSrcBr]	*MacAddr* see 2.296.1.1.1.3	Set source Mac address to router-address.
[metaData] **	*integer*	Set meta data. **Since: 4.25.0
[nextTblName] **	*string*	Next Table Name. **Since: 4.25.0
[outInterfaces]	*list of Interface name*	Output interfaces.
outputAll	*boolean*	Flood to all interfaces except ingress.
outputController	*boolean*	Send to controller.
outputDrop	*boolean*	Drop packet.

outputFlood	*boolean*	Flood to all non-blocked interfaces except ingress.
[outputInternalLoopback] **	*boolean*	Output to hardware loopback. **Since: 4.20.0
outputLocal	*boolean*	Send to switch CPU.
[outputNexthop]	*list of IpGenericAddr* see 2.296.1.1.1.1	Output nexthop IP address.
[outputNexthopRec]	*boolean*	Recursive resolution of nexthop IP address.
[outputNexthopVrfName] **	*string*	VRF name. **Since: 4.22.1
outputNormal	*boolean*	Process normally.
[outputQ]	*integer*	Output queue.
[outputTrafficClass]	*integer*	Output traffic class.
[pktLog] **	*boolean*	Log packet. **Since: 4.26.2
[popVlan] **	*boolean*	Push Vlanid. **Since: 4.23.0
[pushVlan] **	*boolean*	Push Vlanid. **Since: 4.23.0
[removeVlanInner] **	*boolean*	Remove Inner Vlan header. **Since: 4.22.1
[ttlDec]	*boolean*	Decrement TTL.
[vlan]	*integer*	Set Vlan ID.
[vlanPCP]	*integer*	Set Vlan PCP.

2.296.1.1.1 IpGenericAddr

Could also be an instance of:
IpGenericAddrAndPort 2.296.1.1.1.2

ip	*IP address*	IPv4 or IPv6 address.

2.296.1.1.2 IpGenericAddrAndPort

ip	*IP address*	IPv4 or IPv6 address.
port	*integer*	Port.

2.296.1.1.3 MacAddr

mac	*MAC address*	Mac address.

2.296.1.1.4 IpAddr

ip	*IPv4 address*	Ip address.

2.296.1.2 Counter

[matchBytes]	*integer*	Number of bytes that matched flow.
[matchPackets]	*integer*	Number of packets that matched flow.

2.296.1.1.3 Match

[arpSenderProtoAddr] **	*IpAddrAndBitMask* see 2.296.1.1.3.2	Match ARP sender IP address and mask. **Since: 4.28.0
[arpTargetProtoAddr] **	*IpAddrAndBitMask* see 2.296.1.1.3.2	Match ARP target IP address and mask. **Since: 4.28.0
[encapDot1q] **	*boolean*	Match 802.1Q encapsulation. **Since: 4.22.1
[ethType]	*integer*	Match ethernet type.
[ethTypeMask]	*integer*	Match Ethertype Mask.
[fieldSetDst] **	*string*	Match destination field-Set. **Since: 4.22.1
[fieldSetSrc] **	*string*	Match source field-Set. **Since: 4.22.1
[httpMethod] **	*enum: HttpHead* | *HttpDelete* | *HttpPut* | *HttpGet* | *HttpPost* | *HttpOptions* | *HttpPatch*	Match HTTP method. **Since: 4.26.2
[icmpCode] **	*integer*	Match ICMP code. **Since: 4.23.1
[icmpType] **	*integer*	Match ICMP type. **Since: 4.23.1
[inInterfaceCpu]	*boolean*	Match input interface CPU.
[inInterfaces]	*list of Interface name*	Match input interfaces.
[inputInternalLoopback] **	*boolean*	Match hardware loopback. **Since: 4.20.0
[ip6Dst]	*Ip6AddrAndBitMask* see 2.296.1.1.3.1	Match destination IP6 address and mask.
[ip6Src]	*Ip6AddrAndBitMask* see 2.296.1.1.3.1	Match source IP6 address and mask.
[ipDst]	*IpAddrAndBitMask* see 2.296.1.1.3.2	Match destination IP address and mask.
[ipPortDst]	*integer*	Match TCP/UDP destination port.
[ipPortDstMask] **	*integer*	Match TCP/UDP destination port Mask. **Since: 4.27.1
[ipPortSrc]	*integer*	Match TCP/UDP source port.
[ipPortSrcMask] **	*integer*	Match TCP/UDP source port Mask. **Since: 4.27.1
[ipProto]	*integer*	Match IP protocol.
[ipSrc]	*IpAddrAndBitMask* see 2.296.1.1.3.2	Match source IP address and mask.
[ipTos]	*integer*	Match Ip Tos (as received).
[macDst]	*MacAddrAndBitMask* see 2.296.1.1.3.3	Match destination Mac address.
[macDstBr]	*MacAddrAndBitMask* see 2.296.1.1.3.3	Match destination Mac address with router-address.
[macSrc]	*MacAddrAndBitMask* see 2.296.1.1.3.3	Match source Mac address.

[macSrcBr]	*MacAddrAndBitMask* see 2.296.1.1.3.3	Match source Mac address with router-address.
[metaData] **	*integer*	Match metaData. **Since: 4.25.0
[packetRes] **	*enum: L3McKnown | L2BC | L2UcKnown | L3UcUnknown | L3McUnknown | L2McUnknown | L2UcUnknown | L3UcKnown | L2McKnown*	Match Packet Resolution. **Since: 4.18.0
[tcpAck]	*boolean*	Tcp flag Ack.
[tcpFin]	*boolean*	Tcp flag Fin.
[tcpPsh]	*boolean*	Tcp flag Psh.
[tcpRst]	*boolean*	Tcp flag Rst.
[tcpSyn]	*boolean*	Tcp flag Syn.
[tcpUrg]	*boolean*	Tcp flag Urg.
[ttl]	*integer*	Match Time to Live value.
[ttlMask]	*integer*	Match Time to Live mask.
[unknownL2V4Multicast-Address]	*boolean*	Unknown IPV4 multicastaddress. Match on the L2 field.
[unknownL3V4Multicast-Address]	*boolean*	Unknown IPV4 multicastaddress. Match on the L3 field.
[vlan]	*integer*	Match Vlan 'ID.
[vlanInner] **	*integer*	Match inner Vlan 'ID. **Since: 4.22.1
[vlanInnerMask] **	*integer*	Match inner Vlan ID Mask. **Since: 4.22.1
[vlanMask]	*integer*	Match Vlan ID Mask.
[vlanPCP]	*integer*	Match Vlan PCP.
[vlanUnTagged] **	*boolean*	Match untagged packets. **Since: 4.20.0
[vrfName] **	*string*	Match VRF. **Since: 4.23.0

2.296.1.1.3.1 Ip6AddrAndBitMask

ip	*IPv6 address*	Ip6 address.
mask	*IPv6 address*	Ip6 address mask.

2.296.1.1.3.2 IpAddrAndBitMask

ip	*IPv4 address*	Ip address.
mask	*IPv4 address*	Ip address mask.

2.296.1.1.3.3 MacAddrAndBitMask

mac	*MAC address*	Mac address.
mask	*MAC address*	Mac address mask.

2.296.1.1.4 Status

status	*enum: flowRejectedBadMatch | flowRejectedOther | flowRemovedIntfDown | flowRejectedHwTableFull | flowActionsUnsupported | flowRejectedTimeoutNotSupported | flowExpiredHard | flowRejectedBadAction | flowUnknown | flowDeletedGroupRemoved | flowDeleted | flowExpiredIdle | flowCreated | flowPending*	Flow status per slice.

2.297 show directflow detail [...]

revision 2

[details]	*Details* see 2.297.1	Details of flows.
sources	*table(string) = Flows* see 2.296.1	Mapping between Dynamic flow source and Flows.

2.297.1 Details

[numFlowsProgrammed]	*integer*	Number of programmed flows.
sources	*table(string) = FlowStatus* see 2.297.1.1	Mapping between Dynamic flow source and FlowStatus.

2.297.1.1 FlowStatus

status	*table(string) = string*	Mapping of flow name to flow status.

2.298 show directflow field-set

enabled	*boolean*	DirectFlow enabled.
[endPoints]	*DirectFlowEndPointList* see 2.298.1	endpoint list.
fieldSetName	*string*	FieldSet.
status	*enum: success | rejectedOther | failed | pending*	FieldSet status.

2.298.1 DirectFlowEndPointList

endPoints	*list of DirectFlowEndPoint* see 2.298.1.1	endpoints.

2.298.1.1 DirectFlowEndPoint

fieldSetName	*string*	FieldSet.
ipAddr	*IpGenericAddr* see 2.296.1.1.1.1	IP address of endpoint.
macAddr	*MacAddr* see 2.296.1.1.1.3	MAC address of endpoint.
status	*enum: success* | *rejectedOther* | *failed* | *pending*	Endpoint status.
vlanId	*integer*	vlan of endpoint.
[vrfName]	*string*	Vrf Name.

2.299 show directflow flows [...]

revision 2

see 2.297

2.300 show dns cache counters

evictions	*integer*	Evictions.
hits	*integer*	Queries answered locally.
insertions	*integer*	Insertions.
misses	*integer*	Queries forwarded.
size	*integer*	Cache size.

2.301 show dns domain

domainName	*string*	The domain name of the switch.

2.302 show dot1q-tunnel [...]

interfaces	*list of Interface name*	List of interfaces in dot1q-tunnel mode.

2.303 show dot1x all

dynAuth	*boolean*	Dot1X Dynamic Authorization is enabled.
interfaces	*table(Interface name) = Dot1xInterfaceInformation* see 2.303.1	A mapping of interface to its Dot1X information.
lldpBypass	*boolean*	Dot1X LLDP Bypass is enabled.
sendIdentityReqLoggedOffMac **	*boolean*	Send unicast identity request when a MAC is logged off. **Since: 4.27.2
sendIdentityReqNewMac **	*boolean*	Send unicast identity request on a new MAC. **Since: 4.27.2
sessionMoveEapol **	*boolean*	Dot1x EAPOL session move is enabled. **Since: 4.29.0
sessionReplaceDetection-Action **	*enum: logOnly | errdisable | detectionDisabled*	Action to take when frequent session replacements detected. **Since: 4.29.2
sessionReplaceDetection-Enabled **	*boolean*	Detect frequent session replacements in single host mode. **Since: 4.29.1
systemAuthControl	*boolean*	System Authentication Control is enabled.
version	*integer*	Dot1X protocol version.

2.303.1 Dot1xInterfaceInformation

Could also be an instance of:
Dot1xInterfaceDetails 2.303.2

authFailVlan	*integer*	Authentication failure VLAN.
eapolDisabled	*boolean*	EAPOL is disabled.
forceAuthPhone **	*boolean*	Forced authorization of phone devices. **Since: 4.26.2
hostMode	*enum: authAllHost | singleHost | multiHost*	singleHost – Allow only one supplicant on an authorized port. multiHost – Allow multiple clients on an authorized port. authAllHost – Allow multiple dot1x-authenticated clients on an authorized port.
[idleTimeout] **	*integer*	Idle-Timeout for supplicants in seconds. **Since: 4.26.2
maxReauthReq	*integer*	Maximum number of reauthentication attempts.
mbaAuthAlways **	*boolean*	Retain MAC authentication session when EAPOL is in progress. **Since: 4.27.2
mbaEnabled	*boolean*	MAC-based authentication is enabled.
mbaFallback	*boolean*	MBA fallback for EAPOL authentication failure.
mbaHostMode **	*boolean*	MAC based authentication host mode is enabled. **Since: 4.25.1
mbaTimeout	*floating point number*	Timeout for starting MAC based authentication in seconds.

portControl	*enum: forceUnauth* | *controlled* | *forceAuth*	Controlled – enable 802.1X for the interface. ForceAuth – disable 802.1X and put the interface into authorized state. ForceUnauth – disable 802.1X and put the interface into unauthorized state.
quietPeriod	*floating point number*	Waiting period in seconds after a failed authentication.
reauthTimeoutIgnore	*boolean*	Retain current port auth status on reauth server or supplicant timeouts or new reauth requests.
sessionReplaceEnabled **	*boolean*	Dot1x session replace in single-host mode is enabled. **Since: 4.28.1
txPeriod	*floating point number*	Waiting period in seconds before retrying a message to supplicant.
unauthorizedAccessVlanEgress	*boolean*	Give unauthorized access port egress membership of access VLAN.
unauthorizedNativeVlanEgress	*boolean*	Give unauthorized trunk port egress membership of native VLAN.

2.303.2 Dot1xInterfaceDetails

authFailVlan	*integer*	Authentication failure VLAN.
eapolDisabled	*boolean*	EAPOL is disabled.
forceAuthPhone **	*boolean*	Forced authorization of phone devices. **Since: 4.26.2
hostMode	*enum: authAllHost* | *singleHost* | *multiHost*	singleHost – Allow only one supplicant on an authorized port. multiHost – Allow multiple clients on an authorized port. authAllHost – Allow multiple dot1x-authenticated clients on an authorized port.
[idleTimeout] **	*integer*	Idle-Timeout for supplicants in seconds. **Since: 4.26.2
maxReauthReq	*integer*	Maximum number of reauthentication attempts.
mbaAuthAlways **	*boolean*	Retain MAC authentication session when EAPOL is in progress. **Since: 4.27.2
mbaEnabled	*boolean*	MAC-based authentication is enabled.
mbaFallback	*boolean*	MBA fallback for EAPOL authentication failure.
mbaHostMode **	*boolean*	MAC based authentication host mode is enabled. **Since: 4.25.1
mbaTimeout	*floating point number*	Timeout for starting MAC based authentication in seconds.
[portAuthorizationState]	*enum: authorized* | *blocked* | *guestVlan* | *restrictedVlan*	Blocked – port is unauthorized. Authorized – port is authorized. GuestVlan – port is authorized through guest VLANRestrictedVlan – port is authorized through restricted VLAN.

portControl	*enum: forceUnauth | controlled | forceAuth*	Controlled – enable 802.1X for the interface. ForceAuth – disable 802.1X and put the interface into authorized state. ForceUnauth – disable 802.1X and put the interface into unauthorized state.
quietPeriod	*floating point number*	Waiting period in seconds after a failed authentication.
reauthTimeoutIgnore	*boolean*	Retain current port auth status on reauth server or supplicant timeouts or new reauth requests.
sessionReplaceEnabled **	*boolean*	Dot1x session replace in single-host mode is enabled. **Since: 4.28.1
supplicantMacs	*table(MAC address) = integer*	Mapping of supplicant MAC address to its reauthentication period in seconds.
txPeriod	*floating point number*	Waiting period in seconds before retrying a message to supplicant.
unauthorizedAccessVlanEgress	*boolean*	Give unauthorized access port egress membership of access VLAN.
unauthorizedNativeVlanEgress	*boolean*	Give unauthorized trunk port egress membership of native VLAN.

2.304 show dot1x all brief

dynAuth	*boolean*	Dot1X Dynamic Authorization is enabled.
interfaces	*table(Interface name) = Dot1xInterfaceDetails* see 2.303.2	A mapping of interface to its Dot1X detailed information.
lldpBypass	*boolean*	Dot1X LLDP Bypass is enabled.
sendIdentityReqLoggedOffMac **	*boolean*	Send unicast identity request when a MAC is logged off. **Since: 4.27.2
sendIdentityReqNewMac **	*boolean*	Send unicast identity request on a new MAC. **Since: 4.27.2
sessionMoveEapol **	*boolean*	Dot1x EAPOL session move is enabled. **Since: 4.29.0
sessionReplaceDetectionAction **	*enum: logOnly | errdisable | detectionDisabled*	Action to take when frequent session replacements detected. **Since: 4.29.2
sessionReplaceDetectionEnabled **	*boolean*	Detect frequent session replacements in single host mode. **Since: 4.29.1
systemAuthControl	*boolean*	System Authentication Control is enabled.
version	*integer*	Dot1X protocol version.

2.305 show dot1x all details

Could also be an instance of:
Dot1xAllSummary 2.304

dynAuth	*boolean*	Dot1X Dynamic Authorization is enabled.
interfaces	*table(Interface name)* = *Dot1xInterfaceDetails* see 2.303.2	A mapping of interface to its Dot1X detailed information.
lldpBypass	*boolean*	Dot1X LLDP Bypass is enabled.
sendIdentityReqLoggedOffMac **	*boolean*	Send unicast identity request when a MAC is logged off. **Since: 4.27.2
sendIdentityReqNewMac **	*boolean*	Send unicast identity request on a new MAC. **Since: 4.27.2
sessionMoveEapol **	*boolean*	Dot1x EAPOL session move is enabled. **Since: 4.29.0
sessionReplaceDetection-Action **	*enum: logOnly* | *errdisable* | *detectionDisabled*	Action to take when frequent session replacements detected. **Since: 4.29.2
sessionReplaceDetection-Enabled **	*boolean*	Detect frequent session replacements in single host mode. **Since: 4.29.1
systemAuthControl	*boolean*	System Authentication Control is enabled.
version	*integer*	Dot1X protocol version.

see 2.304

2.306 show dot1x all statistics

dropCounters **	*table(Interface name) = Dot1xDroppedCounterStats* see 2.306.1	A mapping between interfaces and 802.1X interface dropped counter statistics. **Since: 4.25.2
interfaces	*table(Interface name) = Dot1xInterfaceStats* see 2.306.2	A mapping between interfaces and 802.1X interface statistics.

2.306.1 Dot1xDroppedCounterStats

unauthEapolHost	*integer*	EAPOL unauthorized host.
unauthEapolPort	*integer*	EAPOL unauthorized port.
unauthMbaHost	*integer*	MBA unauthorized host.

2.306.2 Dot1xInterfaceStats

lastRxSrcMac	*MAC address*	Last EAPOL frame source.
rxInvalid	*integer*	Invalid EAPOL frames received.
rxLogoff	*integer*	EAPOL Logoff frames received.
rxResp	*integer*	EAP Response frames received.
rxRespId	*integer*	EAPOL Resp/Id frames received.
rxStart	*integer*	EAPOL Start frames received.
rxTotal	*integer*	EAPOL frames received.
rxVersion	*integer*	Last EAPOL frame version.
txReq	*integer*	EAP Request frames transmitted.
txReqId	*integer*	EAP Initial Request frames transmitted.
txTotal	*integer*	EAPOL frames transmitted.

2.307 show dot1x all summary

see 2.304

2.308 show dot1x hosts [...]

intfSupplicantsDict	*table(Interface name) = Dot1xHosts* see 2.308.1	A mapping between the interfaces and all the MBA or Dot1X supplicants for that interface.

2.308.1 Dot1xHosts

supplicants	*table(MAC address) = Dot1xHost* see 2.308.1.1	A mapping betweeen MAC addresses and Dot1X or MBA supplicant information.

2.308.1.1 Dot1xHost

authMethod	*string*	Authentication method used by the supplicant.
authStage	*string*	Supplicant state.

cachedAuthAtLinkDown **	*boolean*	Cache supplicant on link down. **Since: 4.24.2
fallback **	*string*	Fallback applied. **Since: 4.25.2
reauthTimeoutSeen **	*boolean*	Reauth timeout seen. **Since: 4.24.2
supplicantMac	*MAC address*	supplicant MAC address.
vlanId	*string*	VLAN ID.

2.309 show dot1x hosts blocked

macAddresses	*table(MAC address) = Dot1xBlockedMacEntry* see 2.309.1	A mapping between blocked MAC address and list of intferace and VLAN pair on which the MAC is blocked through dynamic authorization.

2.309.1 Dot1xBlockedMacEntry

interfaces	*table(Interface name) = integer*	A mapping between interface and VLAN.

2.310 show dot1x hosts mac MAC [...]

accountingSessionId	*string*	Supplicant Accounting-Session-Id.
[aristaPeriodicIdentity] **	*string*	Supplicant Arista-PeriodicIdentity. **Since: 4.27.2
[aristaSegmentId] **	*string*	Supplicant Arista-Segment-Id. **Since: 4.26.2
[aristaWebAuth]	*string*	Supplicant Arista-WebAuth.
authMethod	*string*	Authentication method used by the supplicant.
authStage	*string*	Supplicant state.
cachedAuthAtLinkDown	*boolean*	Cache supplicant on link down.
callingStationId	*string*	Supplicant Calling-Station-Id.
[captivePortal]	*string*	Supplicant Captive portal.
[captivePortalSource]	*string*	Supplicant Captive portal source.
detail_	*boolean*	Print details.
[deviceType] **	*string*	Device Type classification received from AAA. **Since: 4.30.0
fallback **	*string*	Fallback applied. **Since: **currentRelease**
[filterId]	*string*	Supplicant Filter-Id.
[framedIpAddrSource]	*string*	Supplicant Framed-IP-Address source.
[framedIpAddress]	*IPv4 address*	Supplicant Framed-IP-Address.
identity	*string*	EAP identity in use on this interface.
[idleTimeout] **	*integer*	Supplicant Idle-Timeout in seconds. **Since: 4.27.0
interface	*interface name*	Supplicant interface.
[nasFilterRules] **	*list of string*	The list of supplicant NAS-Filter-Rule. **Since: 4.26.2
reauthBehavior	*string*	Supplicant re-authentication behavior.
reauthInterval	*integer*	Supplicant re-authentication interval in seconds.
reauthTimeoutSeen	*boolean*	Reauth timeout seen.
[serviceType]	*string*	Supplicant Service-Type.
[sessionTimeout]	*integer*	Supplicant Session-Timeout.
[supplicantClass]	*string*	Supplicant Class.
supplicantMac	*MAC address*	Supplicant MAC address.
[terminationAction]	*string*	Supplicant Termination-Action.
[timeToReauth] **	*integer*	Time remaining for next reauthentication in seconds. **Since: 4.27.2
[tunnelPrivateGroupId]	*string*	Supplicant Tunnel-Private-GroupId.
vlanId	*string*	VLAN ID.

2.311 show dot1x interface INTFS

interfaces	*table(Interface name) = Dot1xInterfaceInformation* see 2.303.1	A mapping between interfaces and 802.1X interface information.

2.312 show dot1x interface INTFS details

interfaces	*table(Interface name) = Dot1xInterfaceDetails* see 2.303.2	A mapping between interfaces and 802.1X interface detailed information.

2.313 show dot1x interface INTFS statistics

see 2.306

2.314 show dot1x radius

hosts	*table(string) = floating point number*	A mapping of hostname with the timestamp when it is /will be marked as failed.

2.315 show dot1x supplicant [...]

interfaces	*table(Interface name) = Dot1xSupplicant* see 2.315.1	A mapping between interfaces and 802.1X supplicant interface information.

2.315.1 Dot1xSupplicant

authenticatorMac	*MAC address*	802.1X authenticator MAC address for the interface.
eapMethod	*enum: fast | tls | unset*	unset – EAP method is currently unset.fast – EAP method in use is EAP FAST.tls – EAP method in use is TLS.
eapTlsCipher **	*string*	EAP TLS cipher. **Since: 4.27.1
eapTlsVersion **	*string*	EAP TLS version. **Since: 4.27.1
[fipsMode] **	*boolean*	FIPS restrictions enabled. **Since: 4.28.1
identity	*string*	EAP identity in use on this interface.
sslProfileName **	*string*	SSL profile name. **Since: 4.26.0
status	*enum: connecting | down | unused | success | failed*	unused – 802.1X supplicant is uninitializeddown – 802.1X supplicant is down.connecting – 802.1X supplicant is connecting.failed – 802.1X supplicant has failed.success – 802.1X supplicant has succeeded.
supplicantMac	*MAC address*	802.1X supplicant MAC address for the interface.

2.316 show dot1x vlan assignment group [...]

vlanGroupNameDict	*table(string) = string*	A mapping of group name to that group's VLANs.

2.317 show dynamic prefix-list [...]

vrfs	*table(string) = DynamicPrefixLists* see 2.317.1	Dynamic prefix list information for VRFs.

2.317.1 DynamicPrefixLists

dynamicPrefixLists	*table(string) = DynamicPrefixListModel* see 2.317.1.1	Table of dynamic prefix lists by name.

2.317.1.1 DynamicPrefixListModel

[bgpSession] **	*BgpSession* see 2.317.1.1.1	BGP session activated dynamic prefix list information. **Since: 4.30.1
[contributingRoutes] **	*list of DynamicPrefixListContributingRoutes* see 2.317.1.1.2	Capped list of contributing routes (a sample of contributing routes), current cap is 10. **Since: 4.21.0
[ipv4PrefixList]	*string*	IPv4 prefix list name.
[ipv6PrefixList]	*string*	IPv6 prefix list name.
[matchRcfFunction] **	*string*	Match RCF function name. **Since: **currentRelease**
[matchRouteMap]	*string*	Match route map name.
[numContributingRoutes] **	*integer*	Number of contributing routes. **Since: 4.21.0
state	*enum: notApplicableToRoutingInstance | inactive | active*	State of the dynamic prefix list.

2.317.1.1.1 BgpSession

peers	*table(string) = PeerSpecification* see 2.317.1.1.1.1	Peer specifications.

2.317.1.1.1.1 PeerSpecification

originatePeerAddress	*boolean*	Originate peer-address.
sessionState	*enum: OpenConfirm | Active | Unknown | Connect | Idle | Established | OpenSent*	The state of BGP session.

2.317.1.1.2 DynamicPrefixListContributingRoutes

`prefix`	*IP address prefix*	Route Prefix.

2.318 show e-tree vlan

`vlans`	*table(numeric string) = EtreeRoleModel* see 2.318.1	A VLAN's E-Tree role, keyed by VLAN ID.

2.318.1 EtreeRoleModel

`etreeRoleLeaf`	*boolean*	True if this VLAN has the E-Tree leaf role.

2.319 show environment cooling [...]

airflowDirection	*enum: unknownAirflowDirection* | *mismatchedAirflow* | *frontToBackAirflow* | *backToFrontAirflow*	The direction of air flow in the system (front refers to port-side).
[ambientTemperature]	*floating point number*	Ambient temperature detected by the cooling system (in Celsius).
[configuredZones] **	*integer*	Configured number of cooling zones. **Since: 4.27.1
coolingMode	*enum: debug* | *manual* | *automatic*	Mode of operation of the cooling system.
[currentZones] **	*integer*	Current number of cooling zones. **Since: 4.27.1
[defaultZones] **	*boolean*	Default number of cooling zones is used. **Since: 4.27.1
fanTraySlots	*list of FanCollection* see 2.319.1	Fan tray slots in the system.
[minFanSpeed] **	*integer*	User-set minimum fan speed (as %). **Since: 4.22.0
numCoolingZones **	*list of integer*	Supported number of cooling zones. **Since: 4.27.1
[overrideFanSpeed]	*integer*	Fan speed at which manual override is configured (as %).
powerSupplySlots	*list of FanCollection* see 2.319.1	Power supply slots in the system.
shutdownOnInsufficientFans	*boolean*	Shut the system down on insufficient fans if True.
systemStatus	*enum: fanTrayFailedOrMissing* | *incompatibleFans* | *unknownCoolingAlarmLevel* | *coolingOk* | *insufficientFans*	System cooling status.

2.319.1 FanCollection

fans	*list of Fan* see 2.319.1.1	The fans in the fan tray.
label	*string*	Slot Label.
[speed]	*integer*	The aggregated speed of the fans in the tray (as %).
status	*enum: notInserted* | *ok* | *powerLoss* | *unsupported* | *failed* | *unknownHwStatus*	The status of the fan tray.

2.319.1.1 Fan

[actualSpeed]	*integer*	Actual speed of the fan (as %).
[configuredSpeed]	*integer*	Configured speed for the fan (as %).
label	*string*	Label for the fan.

`[lastSpeedStableChangeTime]` **	*floating point number*	Timestamp in UTC (in seconds) when last time the fan's speed was stable. **Since: 4.20.0
`[maxSpeed]` **	*integer*	Maximum speed for the fan (in rpm). **Since: 4.20.0
`[speedHwOverride]` **	*boolean*	If the fan's speed is overridden by hardware. **Since: 4.21.3
`[speedStable]` **	*boolean*	If the fan's speed is stable. **Since: 4.20.0
`status`	*enum: ok* | *powerLoss* | *unsupported* | *failed* | *unknownHwStatus*	Status of the fan.
`[uptime]` **	*floating point number*	Timestamp in UTC (in seconds) when the last change in Fan's status occurred. **Since: 4.20.0

2.320 show environment power

[details]	*Details* see 2.320.1	Details about the power supplies.
powerSupplies	*table(string) = PowerSupply* see 2.320.2	Power supplies status.

2.320.1 Details

pollInterval	*floating point number*	Polling interval (seconds).

2.320.2 PowerSupply

[blackBoxLog] **	*string*	Debugging log. **Since: 4.21.0
[capacity]	*floating point number*	Capacity (watts).
[details]	*boolean*	Details available.
[dominant] **	*boolean*	dominant power supply. **Since: 4.27.2
fans	*table(string) = Fan* see 2.320.2.1	Power supply fans.
[inputBCurrent] **	*floating point number*	Input B current (amps). **Since: 4.22.1
[inputBVoltage] **	*floating point number*	Input B voltage (volts). **Since: 4.22.1
[inputCurrent]	*floating point number*	Input current (amps).
[inputPower]	*floating point number*	Input power (watts).
[inputVoltage]	*floating point number*	Input voltage (volts).
[managed]	*boolean*	Power supply is managed.
[mfrId] **	*string*	Manufacturer ID of power supply. **Since: 4.18.0
[mfrModel] **	*string*	Manufacturer model of power supply. **Since: 4.18.0
[mfrRevision] **	*string*	Manufacturer revision of power supply. **Since: 4.18.0
modelName	*string*	Type of power supply.
[outputCurrent]	*floating point number*	Output current (amps).
[outputPower]	*floating point number*	Output power (watts).
[outputVoltage]	*floating point number*	Output voltage (volts).
[priFirmware] **	*string*	Primary firmware of power supply. **Since: 4.18.0
[secFirmware] **	*string*	Secondary firmware of power supply. **Since: 4.18.0
[specificInfo]	*string*	Power supply model-specific information.
[state]	*enum: ok | failed | unknown | powerLoss*	State of the power supply.
tempSensors	*table(string) = TempSensor* see 2.320.2.2	Power supply temperature sensors.

[uptime] **	*floating point number*	Timestamp in UTC (in seconds) when the last change in PowerSupply's state occurred. **Since: 4.20.0

2.320.2.1 Fan

speed	*integer*	Fan speed (percentage).
status	*enum: ok | failed | unknownHwStatus*	Fan status.

2.320.2.2 TempSensor

status	*enum: ok | disabled | invalidReading | failed | unknownHwStatus*	Temp sensor status.
[temperature]	*floating point number*	Fan temperature (celsius).

2.321 show environment power detail

see 2.320

2.322 show environment power voltage

revision 2

cardSlots	*table(string) = FruSlot* see 2.322.1	A mapping of labels to card slots.
powerSupplySlots	*table(string) = FruSlot* see 2.322.1	A mapping of labels to power supply slots.
voltageSensors	*table(string) = VoltageSensor* see 2.322.1.2	A mapping of labels to voltage sensors.

2.322.1 FruSlot

[currentSensors] **	*table(string) = CurrentSensor* see 2.322.1.1	A mapping of labels to current sensors. **Since: 4.27.1
entPhysicalClass	*string*	Physical entity class of the slot.
label	*string*	Label of the slot.
voltageSensors	*table(string) = VoltageSensor* see 2.322.1.2	A mapping of labels to voltage sensors.

2.322.1.1 CurrentSensor

[current]	*floating point number*	Measured sensor current (amps).
[name]	*string*	The current sensor's hardware label.

`status`	*enum: ok | failed | unknownHwStatus*	Current sensor status.

2.322.1.2 VoltageSensor

`[description]`	*string*	Hardware label describing the voltage sensor.
`[expectedVoltage]`	*floating point number*	Expected sensor voltage (volts).
`status`	*enum: ok | failed | unknownHwStatus*	Voltage sensor status.
`[voltage]`	*floating point number*	Measured sensor voltage (volts).

2.323 show environment temperature [...]

actionOnOverheat **	*enum: actionIgnore* | *actionPowercycle* | *actionShutdown* | *actionUnknown*	System action for an overheat event. **Since: 4.29.0
ambientThreshold **	*integer*	Ambient temperature (in Celsius) above which EOS will run in reduced capacity. **Since: 4.26.2
cardSlots	*list of FruSlot* see 2.323.1	Card slots on a modular system.
powerSupplySlots	*list of FruSlot* see 2.323.1	Power Supply Slots on the system.
powercycleOnOverheat **	*boolean*	Whether the system is configured to power-cycle on an overheat event. **Since: 4.26.0
recoveryModeOnOverheat **	*enum: recoveryModeRestricted* | *recoveryModeNA* | *recoveryModeNormal*	Specify recovery behavior after power-cycle. **Since: 4.26.2
shutdownOnOverheat	*boolean*	Whether the system is configured to shutdown on an overheat event.
systemStatus	*enum: unknown-TemperatureAlarmLevel* | *temperatureCritical* | *temperatureOverheating* | *temperatureOk* | *temperatureSensorFailure*	Temperature status of the system.
tempSensors	*list of TempSensor* see 2.323.1.1	Temperature sensors on a fixed system.

2.323.1 FruSlot

entPhysicalClass	*string*	Physical entity class of the slot.
relPos	*string*	Slot relative position.
tempSensors	*list of TempSensor* see 2.323.1.1	The temperatures sensors in the card slot.

2.323.1.1 TempSensor

alertCount	*integer*	Number of times the alert has been triggered.
[alertLastChange]	*floating point number*	Last time the alert status changed in POSIX time.
[criticalThreshold]	*floating point number*	Configured critical temperature threshold to trigger a system shutdown (in Celsius).
[currentTemperature]	*floating point number*	Current temperature reported by the temperature sensor (in Celsius).
description	*string*	Temperature sensor description.
hwStatus	*enum: ok* | *disabled* | *invalidReading* | *failed* | *unknownHwStatus*	Hardware status reported by the temperature sensor.
inAlertState	*boolean*	Whether the system is in alert state.

isPidDriver **	*boolean*	Whether the sensor is currently the PID driver. **Since: 4.18.0
[lastPidDriverTime] **	*floating point number*	Last time the sensor was the PID driver. **Since: 4.18.0
[maxTemperature]	*floating point number*	Maximum temperature recorded by the temperature sensor (in Celsius).
[maxTemperatureLastChange]	*floating point number*	Last time the maximum temperature recorded by the temperature sensor changed in POSIX time.
name	*string*	Temperature sensor name.
[overheatThreshold]	*floating point number*	Configured temperature threshold to trigger an overheating event (inCelsius).
pidDriverCount **	*integer*	Number of times the sensor has been the PID driver. **Since: 4.18.0
relPos	*string*	Temperature sensor relative position.
[setPointTemperature] **	*floating point number*	Current setpoint temperature used by the temperature sensor (in Celsius). **Since: 4.18.0
[targetTemperature] **	*floating point number*	Configured target temperature used for PID tuning (in Celsius). **Since: 4.18.0

2.324 show environment temperature transceiver [...]

cardSlots	*list of FruSlot* see 2.323.1	Card slots on a modular system.
tempSensors	*list of TempSensor* see 2.323.1.1	Temperature sensors on a fixed system.

2.325 show etherchannel [...]

portChannels	*table(Interface name) = PortChannel* see 2.325.1	Map port-channels to their member information.

2.325.1 PortChannel

activePorts	*table(Interface name) = ActivePort* see 2.325.1.1	Map active ports to time they became active, protocol, and mode.
[currWeight] **	*integer*	Port channel active ports total weight. **Since: 4.26.2
[fallbackState]	*enum: unconfigured | configured | locallyOnly | monitoring | passive | enabled | initialized*	Port Channel fallback state.
inactiveLag **	*boolean*	Exceeds the port channel config limit. **Since: 4.25.0
inactivePorts	*table(Interface name) = InactivePort* see 2.325.1.2	Map inactive ports to time and reason they became inactive.
[maxWeight] **	*integer*	Port channel active ports maximum total weight allowed. **Since: 4.26.2
[minLinks] **	*integer*	Port channel configured minimum links. **Since: 4.26.2
[minSpeed] **	*string*	Port channel configured minimum speed (gbps or mbps). **Since: 4.26.2
recircFeature	*list of string*	Contains a list of features that uses recirc-channel.
rxPorts **	*table(Interface name) = RxPort* see 2.325.1.3	Map receive-only ports to time they start receiving and reason they don't transmit. **Since: 4.27.0

2.325.1.1 ActivePort

[lacpMode]	*enum: off | passive | active*	LACP active or passive mode.
[protocol]	*enum: lacp | unconfigured | static*	Lag mode configuration.
[timeBecameActive]	*floating point number*	UTC time the port became active.
[weight] **	*integer*	Member port weight. **Since: 4.26.2

2.325.1.2 InactivePort

[reasonUnconfigured]	*string*	Reason interface is not part of the LAG.
[timeBecameInactive]	*floating point number*	UTC time the interface was no longer active.

2.325.1.3 RxPort

[reason]	*string*	Reason port is receiving but not transmitting.
[time]	*floating point number*	UTC time the port started receiving.

2.326 show event-handler [...]

revision 2

eventHandlers	*table(string) = EventHandler* see 2.326.1	A mapping between a handler and hander information.

2.326.1 EventHandler

action	*string*	The action to take when a trigger occurs.
actionCount	*integer*	The number of actions that have occured.
actionKind **	*string*	Type of action to take when a trigger occurs. **Since: 4.22.1
asynchronous	*boolean*	Indicates if action is non-blocking.
builtinHandler	*boolean*	Indicates that this handler is built-in, not user configured.
builtinHandlerDisabled	*boolean*	Indicates that this built-in handler ismanually disabled by the user.
delay	*floating point number*	The configured event-handler delay in seconds.
detectionCount	*integer*	The number of trigger detections that have occured.
eventSubhandlers **	*table(string) = EventSubhandler* see 2.326.1.1	A mapping between a subhandler for this handler and its status information. **Since: 4.22.1
hasSubhandlers **	*boolean*	The handler has per-source granularity. **Since: 4.22.1
[healthAction]	*string*	The health action to take when a trigger occurs.
[lastActionTime]	*floating point number*	UTC timestamp of last action.
[lastDetectionTime]	*floating point number*	UTC timestamp of last trigger detection.
[lastTriggerTime]	*floating point number*	UTC timestamp of last trigger activation.
maxActionDuration	*floating point number*	The maximum time in seconds the action is allowed to execute before being aborted.
maxRepeatActionCount **	*integer*	Maximum actions permitted in the repeat interval window. **Since: 4.30.0
pollingCount	*integer*	The number of times the event-handler has been polled.
repeatInterval **	*floating point number*	The period in seconds for which repeating events are ignored. **Since: 4.24.2
runUponConfig **	*boolean*	Indicates if action is to carried out when this handler is configured. **Since: 4.26.0
thresholdCount	*integer*	The number of events that must occur within the threshold window.
thresholdWindow	*integer*	The window in seconds where the number of events must match or exceed the threshold count.

[trigger]	*Trigger* see 2.326.1.2	The condition that causes a trigger.
triggerCount	*integer*	The number of triggers that have occured.

2.326.1.1 EventSubhandler

actionCount	*integer*	The number of actions that have occured.
detectionCount	*integer*	The number of trigger detections that have occured.
[lastActionTime]	*floating point number*	UTC timestamp of last action.
[lastDetectionTime]	*floating point number*	UTC timetamp of last trigger detection.
[lastTriggerTime]	*floating point number*	UTC timestamp of last trigger activation.
pollingCount	*integer*	The number of times the event-handler has been polled.
triggerCount	*integer*	The number of triggers that have occured.

2.326.1.2 Trigger

[onCounters]	*OnCounters* see 2.326.1.2.1	Trigger on counters.
[onIntf]	*OnIntf* see 2.326.1.2.2	Trigger on interface.
[onLogging]	*OnLogging* see 2.326.1.2.3	Trigger on logging.
[onMaintenance]	*OnMaintenance* see 2.326.1.2.4	Trigger on maintenance.
triggerType	*enum: onStartupConfig* | *onMaintenance* | *vmTracerVm* | *onCounters* | *onLogging* | *onIntf* | *onBoot*	Configured event trigger condition.

2.326.1.2.1 OnCounters

condition	*string*	The counter condition expression.
pollInterval	*floating point number*	The counter polling interval in seconds.

2.326.1.2.2 OnIntf

interface	*list of Interface name*	List of monitored interfaces.
onChangeOf	*list of string*	Action is triggered upon these changes to the interface.

2.326.1.2.3 OnLogging

logRegex	*string*	Regular expression to use for searching log messages.
pollInterval	*floating point number*	The poll interval in seconds.

2.326.1.2.4 OnMaintenance

[bgp]	*Bgp* see 2.326.1.2.4.1	BGP peer whose maintenance causes a trigger.
[interface]	*interface name*	Name of interface.
onMaintenanceType	*enum: intf | bgp | unit*	The type of maintenance interface.
onMode	*enum: enter | exit*	Maintenance operation that causes a trigger.
onStage	*enum: all | afterLinkdown | beforeLinkdown | beforeBgp | end | afterMlag | afterBgp | beforeRatemon | afterRatemon | begin | beforeMlag*	Trigger condition occurs on maintenance stage.
[unit]	*string*	The unit whose maintenance causes a trigger.

2.326.1.2.4.1 Bgp

[groupName]	*string*	Name of peer group.
[ip4Neighbor]	*IPv4 address*	Neighbor address.
[ip6Neighbor]	*IPv6 address*	Neighbor address.
[vrfName]	*string*	Configured BGP VRF.

2.327 show event-handler trigger counters [...]

[eventHandlerTrigger-CounterSources]	*list of string*	The list of counter sources supported by the on-counters event handler.
[eventHandlerTrigger-Counters]	*table(string) = Event-HandlerTriggerCounters* see 2.327.1	A mapping of counter sources to counters.

2.327.1 EventHandlerTriggerCounters

counters	*list of string*	The list of counters supported by this on-counters event handler counter source.

2.328 show event-monitor lacp [...]

eventMonReasons	*list of LacpEventMon-ReasonDetails* see 2.328.1	LACP event monitor reason.

2.328.1 LacpEventMonReasonDetails

lag	*interface name*	Port channel name.
member	*interface name*	Port channel member interface name.
reason	*string*	Reason for the port channel member state transition.
state	*string*	Port Channel member state.

`time`	*floating point number*	UTC timestamp of the state transition.

2.329 show extensions [...]

revision 2

`extensionStoredDir` **	*string*	The filesystem that extensions are actually stored in. **Since: 4.24.1
`extensions`	*table(string) = Extension* see 2.329.1	Maps filenames to extension details.

2.329.1 Extension

`[affectedAgents]` **	*list of string*	Agents that are affected by this extension in the sence that they need to be restarted for the extension to take effect (static info). **Since: 4.25.1
`[agentsToRestart]` **	*list of string*	Agents that are pending a restart to finalize the extension; will be reset after restart. **Since: 4.25.1
`boot` **	*boolean*	Install at boot. **Since: 4.29.2
`[description]`	*string*	Description to the extension.
`error`	*boolean*	Indicates that the extension is in an error state, for instance if the system believes the extension was installed but one or more of its packages got manually uninstalled.
`[installedSize]`	*integer*	Installed size (in bytes) of the extension.
`[numPackages]`	*integer*	How many packages are installed.
`[packages]`	*table(string) = PackageInfo* see 2.329.1.1	Info about the extension keyed by names of the extensions.
`presence`	*enum: absent | present*	Whether or not the extension is still present on the switch. An extension can be installed and then removed, which would cause it to remain installed and become absent.
`[presenceDetail]`	*string*	Contains an error string explaining the presence of the extension. For instance the extension may be corrupted or otherwise unusable, which would cause the system to consider the extension is absent because it cannot be used.
`release`	*string*	Release of the extension.
`[signed]` **	*boolean*	The extension has a valid signature. **Since: 4.24.0
`[signedDetail]` **	*string*	Contains an error string explaining why the extension is signed incorrectly. **Since: 4.24.0
`status`	*enum: forceInstalled | installed | notInstalled*	Whether the extension is installed.

[statusDetail]	*string*	Contains an error string explaining the status of the extensions. For instance the extension may have failed to be (un)installed on the system, and this string would contain additional information explaining why.
[summary]	*string*	Summary to this extension.
[vendor]	*string*	Vendor of this extension.
version	*string*	Version of the extension.

2.329.1.1 PackageInfo

release	*string*	Release of this particular extension.
version	*string*	Version of this particular extension.

2.330 show feature-toggle

toggleInfo	*table(string) = ArToggleEntry* see 2.330.1	Collection of feature toggles.

2.330.1 ArToggleEntry

default	*integer*	Is this toggle enabled by default.
enabled	*integer*	Is this toggle enabled.

2.331 show feature-toggle override status

name	*string*	Describes this status attribute.
status	*enum: OK | NA | ERROR*	Status of toggle override file parse.

2.332 show file digest HASHTYPE URL

[hashType]	*enum: md5 | sha512*	Hash name which is specified by the given flag.
[hashValue]	*string*	Hash value for the given file path.
path	*string*	File path for which this information is under verification.

2.333 show file information URL

[fileName]	*string*	File name if path is a local file else empty.
[fileType]	*enum: swix | swi | elf | zip | pythonScript | script*	File types are: elf: ELF binary swi: Arista software image with version in version field swix: Arista EOS extension zip: Zip archive data script: A script file pythonScript: A Python script file.

isDir	*boolean*	When this flag is set to True it signifiesthat the given path is a directory. All other fields are empty and not relevant when this flag is set.
[isEmpty]	*boolean*	A flag when set to True signifies that the given path is an empty file.
[isExecutable]	*boolean*	A flag when set to True signifies that the given file path is executable.
path	*string*	Path name for which this information is generated.
[version]	*string*	When fileType is swi this field is the Arista EOS version.

2.334 show file systems

fileSystems	*list of FileSystem* see 2.334.1	List of file systems.

2.334.1 FileSystem

currentFs	*boolean*	When set it signifies this file system is the current one.
free	*integer*	Free space in the file system in KB.
fsType	*enum: flash* | *simple* | *checkpoint* | *network* | *system* | *ssh* | *session* | *alert-base* | *ssl* | *extension* | *peer*	file system type.
[linuxFs] **	*enum: vfat* | *ext4*	Actual filesystem type used on the physical medium. It will be omitted if there's no physical medium attached to the filesystem. **Since: 4.20.5
permission	*enum: rw* | *r* | *w*	File permissions of read-only, write-only and read-write.
prefix	*enum: ssh-ca-key:* | *extension:* | *platform:* | *alert-base:* | *https:* | *crash:* | *ftp:* | *certificate:* | *scp:* | *usb2:* | *nvme0:* | *session:* | *ssh-host-cert:* | *nvme1:* | *flash:* | *drive:* | *ssh-auth-principals-cmd:* | *drive2:* | *http:* | *file:* | *usb1:* | *simple:* | *supervisor-peer:* | *system:* | *checkpoint:* | *tftp:* | *sftp:* | *sslkey:* | *ssh-revoke-list:*	file system prefix.
size	*integer*	Size of the file system in KB.

2.335 show flow telemetry ipfix counters

exporters	*list of IpfixExporterCounter* see 2.335.1	Counters for all exporters.
[receivedMessages]	*integer*	Number of IPFIX/sFlow/Postcard messages received.
[sentMessages]	*integer*	Number of flow information sent.

2.335.1 IpfixExporterCounter

exporterId	*IpfixExporterId* see 2.335.1.1	Exporter ID.
interpretFieldErrors	*integer*	Interpret data record field errors.
invalidFieldErrors	*integer*	Invalid data record field errors.
invalidFlowKeyErrors **	*integer*	Invalid flow key in data record errors. **Since: 4.23.1
invalidMsgHdrErrors	*integer*	Invalid message header errors.
invalidOtrErrors	*integer*	Invalid options template record errors.
invalidSetHdrErrors	*integer*	Invalid set header errors.
invalidSetIdErrors	*integer*	Invalid set ID errors.
invalidSetLenErrors	*integer*	Invalid set length errors.
invalidTrErrors	*integer*	Invalid template record errors.
receivedDrs	*integer*	Received IPFIX data records or sFlow/Postcard flow samples.
receivedMessages	*integer*	Received messages.
receivedOdrs	*integer*	Received options data records.
receivedOtrs	*integer*	Received options template records.
receivedTrs	*integer*	Received template records.
unknownTidErrors	*integer*	Unknown template ID errors.
unsupportedOdrErrors	*integer*	Unsupported options data record errors.

2.335.1.1 IpfixExporterId

address	*IP address*	Exporter IP address.
observationDomainId	*integer*	Observation domain ID.
protocol	*enum: sflow | greent | ipfix*	Exporter flow exporting protocol.
sourcePort	*integer*	Exporter source port number.

2.336 show flow telemetry ipfix flow-table [...]

exporters	*list of IpfixExporterFlow* see 2.336.1	Flows for all exporters.

2.336.1 IpfixExporterFlow

exporterId	*IpfixExporterId* see 2.335.1.1	Exporter ID.
v4Flows	*list of IpfixFlow* see 2.336.1.1	IP flows for this exporter.

v6Flows	*list of IpfixFlow* see 2.336.1.1	IPv6 flows for this exporter.

2.336.1.1 IpfixFlow

flowData	*table(string) = string*	Mapping of Information Element name to value in Flow Data part.
flowKey	*table(string) = string*	Mapping of Information Element name to value in Flow Key part.
[intNodes] **	*list of IpfixIntNode* see 2.336.1.1.1	All Inband Telemetry path nodes for this flow. **Since: 4.24.0
lastUpdateTime	*integer*	UTC timestamp in nano second of the last hostname resolution.

2.336.1.1.1 IpfixIntNode

intIntervals	*list of IpfixIntInterval* see 2.336.1.1.1.1	All intervals for this Inband Telemetry path node.
nodeData	*table(string) = string*	Mapping of Information Element name to Information Element value for an Inband Telemetry path node.

2.336.1.1.1.1 IpfixIntInterval

intervalData	*table(string) = string*	Mapping of Information Element name to Information Element value for an interval in an Inband Telemetry path node.

2.337 show flow telemetry ipfix hostname

v4Hostnames	*list of IpfixHostname* see 2.337.1	Hostname information for all IP addresses.
v6Hostnames	*list of IpfixHostname* see 2.337.1	Hostname information for all IPv6 addresses.

2.337.1 IpfixHostname

address	*IP address*	IP address.
hostnames	*list of string*	Host names of this address.
lastUpdateTime	*integer*	UTC timestamp in nano second of the last hostname resolution.

2.338 show flow telemetry ipfix option-data

exporters	*list of IpfixExporterOptionData* see 2.338.1	Option data for all exporters.

2.338.1 IpfixExporterOptionData

exporterId	*IpfixExporterId* see 2.335.1.1	Exporter ID.
interfaces	*table(string) = string*	Interface ID to interface name mapping.
observationDomains	*table(string) = IpfixDataFields* see 2.338.1.1	Observation domain options data.
templates	*table(string) = IpfixDataFields* see 2.338.1.1	Template options data.
vrfs	*table(string) = string*	VRF ID to interface name mapping.

2.338.1.1 IpfixDataFields

fields	*table(string) = string*	Information Element name to value mapping.

2.339 show flow telemetry ipfix template

exporters	*list of IpfixExporterTemplate* see 2.339.1	Templates for all exporters.

2.339.1 IpfixExporterTemplate

exporterId	*IpfixExporterId* see 2.335.1.1	Exporter ID.
templateInfos	*list of IpfixTemplateInfo* see 2.339.1.1	Templates for this exporter.

2.339.1.1 IpfixTemplateInfo

fields	*list of IpfixTemplateField* see 2.339.1.1.1	List of fields.
scopeFields	*list of IpfixTemplateField* see 2.339.1.1.1	List of scope fields.
templateId	*integer*	Template ID.
timeReceived	*integer*	Unix epoch time when template was last received.

2.339.1.1.1 IpfixTemplateField

enterpriseNumber	*integer*	IANA Enterprise number.
informationElementId	*integer*	Information Element ID.
informationElementName	*string*	Information Element name.
length	*integer*	Field length.

2.340 show flow telemetry postcard counters

see 2.335

2.341 show flow telemetry postcard flow-table [...]

see 2.336

2.342 show flow telemetry postcard hostname

see 2.337

2.343 show flow telemetry postcard option-data

see 2.338

2.344 show flow telemetry sflow counters

see 2.335

2.345 show flow telemetry sflow flow-table [...]

see 2.336

2.346 show flow telemetry sflow hostname

see 2.337

2.347 show flow telemetry sflow option-data

see 2.338

2.348 show flow tracking (sampled | hardware | (telemetry inband)) ipfix options-table [...]

`trackers`	*table(string) = TrackerOptTableModel* see 2.348.1	A mapping between tracker name and tracker options table.

2.348.1 TrackerOptTableModel

[tables]	*table(string) = OptTableModel* see 2.348.1.1	A mapping between table name and options table.
[trackerInfo]	*TrackerInfoModel* see 2.348.1.2	Tracker info options table.

2.348.1.1 OptTableModel

entries	*table(numeric string) = string*	A mapping between table entry ID and entry value.
scopeFields	*list of string*	List of options table scope fields.
templateId	*integer*	Options template ID.

2.348.1.2 TrackerInfoModel

activeInterval	*integer*	Flow active interval in seconds.
flowTrackingType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracking type.
inactiveTimeout	*integer*	Flow inactive timeout in seconds.
observationDomain	*string*	Observation domain name.
observationDomainId	*integer*	Observation domain ID.
sampleRate	*integer*	Flow tracking sample every x packets.
selectorAlgorithm	*randomNofNSampling*	Flow selector algorithm.

2.349 show flow tracking (sampled | hardware | (telemetry inband)) ipfix template [...]

trackers	*table(string) = TrackerTemplateModel* see 2.349.1	A mapping between tracker name and tracker templates.

2.349.1 TrackerTemplateModel

dataTemplates	*table(numeric string) = TemplateModel* see 2.349.1.1	A mapping between template ID and data template.
optionsTemplates	*table(numeric string) = TemplateModel* see 2.349.1.1	A mapping between template ID and options template.

2.349.1.1 TemplateModel

fields	*list of TemplateEntryModel* see 2.349.1.1.1	Ordered list of template fields.
[groups] **	*list of string*	List of flow groups using the template. **Since: 4.28.1
name	*string*	Template name.

[scopeFields]	*list of TemplateEntryModel* see 2.349.1.1.1	Ordered list of template scope fields.
templateId	*integer*	Template ID.

2.349.1.1.1 TemplateEntryModel

[enterpriseIe]	*boolean*	IPFIX template filed is enterprise IE.

`fieldId`	*enum: srcIntf* | *samplingPopulation* | *inbandTelemetryLastPacket-Latency* | *aristaInternalVxlanV6-Protocol* | *flowEndReason* | *dstIpv6PrefixLen* | *inbandTelemetryPath-NumberOfBytes* | *timeStampFlowStart* | *inbandTelemetryDeviceId* | *mplsLabelStackSection3* | *mplsTopLabelPrefixLength* | *dot1qVlanId* | *observationDomainName* | *innerSrcL4Port* | *vrfName* | *inbandTelemetry-AggregationInterval* | *vxlanNetworkId* | *inbandTelemetryNode-AverageLatency* | *mplsTopLabelIPv4Address* | *mplsLabelStackSection9* | *ip6NextHop* | *inbandTelemetryNode-MaximumLatency* | *aristaDpsPathLatency* | *inbandTelemetryNode-MinimumLatency* | *applicationId* | *opaqueOctets* | *inbandTelemetryInterval-NumberOfBytes* | *dstIp* | *inbandTelemetryInterval-NumberOfPackets* | *inbandTelemetryTTL* | *aristaDpsPathId* | *selectorAlgorithm* | *dstMac* | *aristaInternalV6Protocol* | *mplsTopLabelType* | *flowDirection* | *mplsLabelStackSection10* | *layer2SegmentId* | *mplsLabelStackSection6* | *srcVlan* | *paddingOctets* | *egressVrfId* | *ipProtocol* | *dstIp6* | *aristaAvtId* | *aristaBscanExportReason* | *mplsLabelStackSection4* | *inbandTelemetryLast-IntervalExportedTimestamp* | *innerSrcIp6* | *flowActiveTimeout* | *bgpDstAs* | *flowEndSeconds* | *dstIntf* | *ingressInterface* | *mplsLabelStackSection7* | *observationDomainId* | *flowKeyIndicator* | *bgpIp6NextHop* | *egressInterface* | *flowIdleTimeout* |	IPFIX template field ID.

`fieldLen`	*integer*	IPFIX template field length (in bytes).
`fieldType`	*string*	IPFIX template field data type.

2.350 show flow tracking (sampled | hardware | mirror-on-drop | (telemetry inband)) [...]

revision 2

[enabledBy]	*string*	The command that enables flow tracking.
[encapTypes]	*list of EncapTypeModel* see 2.350.1	List of encapsulations.
[ftrTrafficGroups]	*table(string) = FtrTrafficTypeModel* see 2.350.3	Flow tracking traffic types.
ftrType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracking type.
running	*boolean*	Flow tracking is running.
[sampleLimit] **	*integer*	Maximum sample rate in packets per second per interface. **Since: 4.30.1
[standardFormatForCounters]	*boolean*	Flow tracking standard record format for counters.
[standardFormatForTimeStamps]	*boolean*	Flow tracking standard record format for timestamps.
trackers	*table(string) = FtModel* see 2.350.2	A mapping between tracker name and tracker.

2.350.1 EncapTypeModel

encapType	*enum: encapIpv6 | encapIpv4 | encapNone | encapMpls | encapVxlan*	Encapsulation.

2.350.2 FtModel

active	*boolean*	Tracker is active.
activeEgressIntfs	*list of IntfModel* see 2.350.2.3	Active Egress Interfaces.
activeInterval	*integer*	Active interval in ms.
activeIntfs	*list of IntfModel* see 2.350.2.3	Active Ingress Interfaces.
exporters	*table(string) = ExpModel* see 2.350.2.1	A mapping between exporter name and exporter.
groups	*table(string) = FgModel* see 2.350.2.2	Flow groups.
inactiveEgressIntfs	*list of IntfModel* see 2.350.2.3	Inactive Egress Interfaces.
inactiveIntfs	*list of IntfModel* see 2.350.2.3	Inactive Ingress Interfaces.
[inactiveReason]	*enum: trackerInactiveReasonUnknown | trackerNotProcessed | maxTrackersLimit*	Tracker inactive reason.
inactiveTimeout	*integer*	Inactive timeout in ms.

2.350.2.1 ExpModel

active	*boolean*	Exporter is ready for transmission.
collectors	*table(string) = CollectorModel* see 2.350.2.1.1	A mapping between collector address and collector port.
[dscpValue]	*integer*	DSCP value in IP header.
exportFormat	*ExpFormatModel* see 2.350.2.1.2	Export format information.
[inactiveReasons]	*table(string) = ExpReasonModel* see 2.350.2.1.3	A mapping between type of reason and list of reasons for inactive exporter.
localIntf	*interface name*	Local interface to be used for IPFIX source IP4/6 addr.
srcIp6Addr	*IPv6 address*	Local IPv6 address to be used for IPFIX source addr.
srcIpAddr	*IPv4 address*	Local IPv4 address to be used for IPFIX source addr.
[templateInterval]	*integer*	Template interval in ms.
[useSflowCollectorConfig]	*boolean*	Use the configured sFlow collectors.
[vrfName]	*string*	VRF name.

2.350.2.1.1 CollectorModel

active	*boolean*	Collector is ready for transmission.
[inactiveReason]	*enum: maxCollectorsLimit | collectorNotProcessed | collectorInactiveReason-Unknown*	Collector inactive reason.
port	*integer*	Collector port.

2.350.2.1.2 ExpFormatModel

[mtu]	*integer*	MTU of the export format.
name	*string*	Export format name.
[version]	*integer*	Version of the export format.

2.350.2.1.3 ExpReasonModel

expReasons	*list of ExpReason* see 2.350.2.1.3.1	List of reasons for inactive exporter.

2.350.2.1.3.1 ExpReason

expReason	*enum: exporterInactive-ReasonUnknown | exporterNotProcessed | invalidVrf | invalidCollector | invalidExportFormat | waitingForUDPPort | maxExportersLimit | invalidLocalIntf*	Exporter inactive reason.

2.350.2.2 FgModel

[encapTypes]	*list of EncapTypeModel* see 2.350.1	List of encapsulations.
[expNames]	*table(string) = GroupExpModel* see 2.350.2.2.2	Exporters, keyed by exporter name.
[hwGroups]	*table(numeric string) = HwFgModel* see 2.350.2.2.3	Flow groups programmed in hardware, keyed by their sequence number.
[ip6AccessLists]	*table(numeric string) = AccessListModel* see 2.350.2.2.1	IPv6 ACLs, keyed by their sequence number.
[ipAccessLists]	*table(numeric string) = AccessListModel* see 2.350.2.2.1	IPv4 ACLs, keyed by their sequence number.
[mirroring]	*MirrorModel* see 2.350.2.2.4	Mirroring details.
[seqno]	*integer*	Flow group sequence number.

2.350.2.2.1 AccessListModel

aclName	*string*	Access list name.

2.350.2.2.2 GroupExpModel

expName	*string*	Exporter name.

2.350.2.2.3 HwFgModel

[active]	*boolean*	Hardware flow group is active.
hwFgName	*string*	Hardware flow group name.

2.350.2.2.4 MirrorModel

[initialCopy]	*integer*	Initial packets to be mirrored.
[mirrorInterval]	*integer*	Sampling interval for packets to be mirrored.
[mirrorSelectorAlgorithm]	*enum: random | sampleModeNone | fixed*	Mirror sampling algorithm.

sessionName	*string*	Mirror session name.

2.350.2.3 IntfModel

interface	*interface name*	Name of the interface.
[reason]	*string*	Reason for interface inactive.

2.350.3 FtrTrafficTypeModel

[hwOffloadStatus]	*FtrHoStatusModel* see 2.350.3.1	Flow tracking hardware offload status.
[sampleRate]	*integer*	Flow tracking sample every x packets.

2.350.3.1 FtrHoStatusModel

[inactiveReasons]	*list of HoInactiveReasonModel* see 2.350.3.1.1	List of hardware offload inactive reasons.
[status]	*enum: unconfigured* | *inactive* | *active*	Flow tracking hardware offload status.

2.350.3.1.1 HoInactiveReasonModel

inactiveReason	*enum: invalidTcamProfile* | *counterFeatureDisabled*	Hardware offload inactive reason.

2.351 show flow tracking hardware counters [...]

[activeFlows]	*integer*	Cumulative active flow count.
[clearTime]	*floating point number*	UTC time when total counters were last cleared.
[expiredFlows]	*integer*	Cumulative expired flow count.
[flows]	*integer*	Cumulative flow count.
[packets]	*integer*	Cumulative count of packets.
running	*boolean*	Flow tracking agent is running.
softwareFlowTable **	*boolean*	Software flow table is available. **Since: 4.23.2
trackers	*table(string) = TrackerCounters* see 2.351.1	A mapping of tracker name to counters.

2.351.1 TrackerCounters

[activeFlows]	*integer*	Number of active flows.
[clearTime]	*floating point number*	UTC time when counters were last cleared.
[expiredFlows]	*integer*	Cumulative expired flow count.
exporters	*table(string) = ExporterCounters* see 2.351.1.1	A mapping of exporter name to counters.
flowGroups	*table(string) = FlowGroupCounters* see 2.351.1.2	A mapping of flow group name to counters.
[flows]	*integer*	Cumulative flow count.
[packets]	*integer*	Cumulative count of packets.

2.351.1.1 ExporterCounters

[clearTime]	*floating point number*	UTC time when counters were last cleared.
collectors	*list of CollectorCounters* see 2.351.1.1.1	A list of per-collector counters.
exporterType	*ipfix*	Exporter type.

2.351.1.1.1 CollectorCounters

addrAndPort	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Collector IP address and port number.
[exportedFlowRecordTotalCount]	*integer*	The total number of flow records sent to the collector.
[exportedMessageTotalCount]	*integer*	The total number of messages sent to the collector.
[exportedOctetTotalCount]	*integer*	The total number of bytes sent to the collector.
lastUpdates	*CollectorTimestamp* see 2.351.1.1.1.3	UTC time of last events.
sets	*table(numeric string) = CollectorSetCounters* see 2.351.1.1.1.1	A mapping of IPFIX template ID (set ID) to per-set-id counters.

templates	*list of CollectorTemplateCounters* see 2.351.1.1.1.2	A list of per-template counters.

2.351.1.1.1.1 CollectorSetCounters

flowRecords	*integer*	Number of flow records sent to the collector.

2.351.1.1.1.2 CollectorTemplateCounters

templateType	*enum: template | optionsTemplate*	IPFIX template type.
templates	*integer*	Number of templates sent to the collector.

2.351.1.1.1.3 CollectorTimestamp

[dataRecord]	*floating point number*	UTC time of last exported data record.
[message]	*floating point number*	UTC time of last exported message.
[optionsData]	*floating point number*	UTC time of last exported options data record.
[template]	*floating point number*	UTC time of last exported template.

2.351.1.2 FlowGroupCounters

activeFlows	*integer*	Number of active flows.
expiredFlows	*integer*	Cumulative expired flow count.
flows	*integer*	Cumulative flow count.
packets	*integer*	Cumulative count of packets.

2.352 show flow tracking hardware flow-table [...]

running	*boolean*	Flow tracking is active.
softwareFlowTable **	*boolean*	Software flow table is available. **Since: 4.23.2
trackers	*table(string) = TrackerModel* see 2.352.1	A mapping between tracker name and tracker.

2.352.1 TrackerModel

groups	*table(string) = GroupModel* see 2.352.1.1	A mapping between hardware group name and group.
numFlows	*integer*	Total number of flows.

2.352.1.1 GroupModel

flows	*list of FlowModel* see 2.352.1.1.1	List of flows.

2.352.1.1.1 FlowModel

bytesReceived	*integer*	Number of bytes received.
[flowDetail]	*FlowDetailModel* see 2.352.1.1.1.1	Flow detailed information.
key	*FlowKeyModel* see 2.352.1.1.1.2	Flow key.
pktsReceived	*integer*	Number of packets received.
startTime	*floating point number*	Flow start time.

2.352.1.1.1.1 FlowDetailModel

dot1qVlanId	*integer*	Flow Dot1q VLAN ID.
[hwFgId] **	*integer*	Hardware flow group ID. **Since: 4.25.1
lastPktTime	*floating point number*	Last packet received time.
tcpFlags	*string*	Flow TCP flags.
[vlanId] **	*integer*	Flow ingress VLAN ID. **Since: 4.25.0

2.352.1.1.1.2 FlowKeyModel

dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.

[ipProtocol]	*enum: ipProtoExperiment1* | *ipProtoRsvp* | *ipProtoL2tp* | *ipProtoIpv6NoNxt* | *ipProtoIpip6* | *ipProtoIcmp* | *ipProtoAh* | *ipProtoPim* | *ipProtoIpv6HopByHop* | *ipProtoTcp* | *ipProtoUdp* | *ipProtoIpip* | *ipProtoIgmp* | *ipProtoGre* | *ipProtoIpv6Route* | *ipProtoSctp* | *ipProtoIcmpv6* | *ipProtoIsis* | *ipProtoOspf* | *ipProtoUnknown* | *ipProtoIpv6Frag* | *ipProtoIpv6Opts* | *ipProtoExperiment2* | *ipProtoEsp* | *ipProtoVrrp*	IP protocol.
ipProtocolNumber	*integer*	IP protocol number.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.
vrfName	*string*	VRF name.

2.353 show flow tracking hardware flow-table [...]_1

running	*boolean*	Flow tracking is active.
softwareFlowTable	*boolean*	Software flow table is available.
trackers	*table(string) = TrackerModel* see 2.353.1	A mapping between tracker name and tracker.

2.353.1 TrackerModel

groups	*table(string) = GroupModel* see 2.353.1.1	A mapping bewteen group name and group.
numFlows	*integer*	Total number of flows.

2.353.1.1 GroupModel

flows	*list of FlowModel* see 2.353.1.1.1	List of flows.

2.353.1.1.1 FlowModel

[appService] **	*string*	Name of service of the application. **Since: 4.28.1
[application] **	*string*	Name of application. **Since: 4.28.1
bytesReceived	*integer*	Number of bytes received.
[flowDetail]	*FlowDetailModel* see 2.353.1.1.1.1	Flow detailed information.
key	*FlowKeyModel* see 2.353.1.1.1.2	Flow key.
pktsReceived	*integer*	Number of packets received.
startTime	*floating point number*	Flow start time.

2.353.1.1.1.1 FlowDetailModel

[appCategory] **	*string*	Name of category of the application. **Since: 4.28.1
[avtName] **	*string*	Name of AVT profile. **Since: 4.28.1
[dpsPathLatency] **	*floating point number*	Latency of DPS path in msecs. **Since: **currentRelease**
[dpsPathLossRate] **	*floating point number*	Loss in DPS path in percentage. **Since: **currentRelease**
[dpsPathName] **	*string*	Name of a DPS path. **Since: 4.28.1
[dpsSourceDevice] **	*boolean*	Device is the source device selecting the DPS path to destination. **Since: **currentRelease**
egressIntf **	*string*	Flow egress interface. **Since: 4.25.2

[ingressIntf] **	*interface name*	Flow ingress interface. **Since: 4.25.2
lastPktTime	*floating point number*	Last packet received time.
tcpFlags	*string*	Flow TCP flags.

2.353.1.1.1.2 FlowKeyModel

dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.
[ipProtocol]	*enum: ipProtoExperiment1 | ipProtoRsvp | ipProtoL2tp | ipProtoIpv6NoNxt | ipProtoIpip6 | ipProtoIcmp | ipProtoAh | ipProtoPim | ipProtoIpv6HopByHop | ipProtoTcp | ipProtoUdp | ipProtoIpip | ipProtoIgmp | ipProtoGre | ipProtoIpv6Route | ipProtoSctp | ipProtoIcmpv6 | ipProtoIsis | ipProtoOspf | ipProtoUnknown | ipProtoIpv6Frag | ipProtoIpv6Opts | ipProtoExperiment2 | ipProtoEsp | ipProtoVrrp*	IP protocol.
ipProtocolNumber	*integer*	IP protocol number.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.
vrfName	*string*	VRF name.

2.354 show flow tracking mirror-on-drop counters [...]

running	*boolean*	Mirror-on-Drop flow tracking is active.
trackers	*table(string) = TrackerModel* see 2.354.1	A mapping between tracker name and tracker information.

2.354.1 TrackerModel

dropReasonCounters **	*table(Interface name) = DropReasonModel* see 2.354.1.1	A mapping between interface and drop reason counters. **Since: 4.30.1
exporters	*table(string) = ExporterModel* see 2.354.1.2	A dictionary of exporters.
groups	*table(string) = GroupModel* see 2.354.1.3	A mapping between group name and group.
intfs	*table(Interface name) = integer*	A mapping between ingress interface and flow count.
numFlows	*integer*	Total number of flows.

2.354.1.1 DropReasonModel

[hardwareError]	*integer*	Hardware error.
[ingressSourcePortFilter]	*integer*	Ingress source port filter.
[ipForwardingLookupMiss]	*integer*	IP forwarding lookup miss.
[ipForwardingLookupMiss-PostMplsDecap]	*integer*	IP forwarding lookup miss after MPLS decap.
[ipMissingArp]	*integer*	ARP unresolved.
[ipMissingArpPostMplsDecap]	*integer*	ARP unresolved after MPLS decap.
[ipMissingRoute]	*integer*	IP missing route.
[ipMissingRoutePostMpls-Decap]	*integer*	IP missing route after MPLS decap.
[ipTtl01]	*integer*	IP TTL 0/1.
[ipTtl01PostMplsDecap]	*integer*	IP TTL 0/1 after MPLS decap.
[ipVersionError]	*integer*	IP version error.
[ipVersionErrorPostMpls-Decap]	*integer*	IP version error after MPLS decap.
[ipv4ChecksumError]	*integer*	IPv4 checksum error.
[ipv4ChecksumErrorPostMpls-Decap]	*integer*	IPv4 checksum error after MPLS decap.
[ipv6MulticastSource]	*integer*	IPv6 multicast source.
[ipv6MulticastSourcePost-MplsDecap]	*integer*	IPv6 multicast source after MPLS decap.
[ipv6Unspecified-Destination]	*integer*	IPv6 unspecified destination.
[ipv6Unspecified-DestinationPostMplsDecap]	*integer*	IPv6 unspecified destination after MPLS decap.
[mplsDisabled]	*integer*	MPLS disabled.
[mplsLabelLookupMiss]	*integer*	MPLS label lookup miss.
[mplsTtl01]	*integer*	MPLS TTL 0/1.

[mtuExceeded]	*integer*	MTU exceeded.
[noForwardingAction]	*integer*	No forwarding action.
[portNotVlanMember]	*integer*	Port not VLAN member.
[rpf]	*integer*	reverse path forwarding.
[saEqualsDa]	*integer*	Source address equals destination address.
[saMulticast]	*integer*	Source address is multicast.
[saNotFound]	*integer*	Source Address not found.
[unknownReason]	*integer*	Unkown reason.

2.354.1.2 ExporterModel

collectors	*table(string) = CollectorModel* see 2.354.1.2.1	A mapping between collector and its statistics.

2.354.1.2.1 CollectorModel

addrAndPort	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Collector IP address and port number.
byteCount	*integer*	Total bytes sent.
packetCount	*integer*	Total packets sent.
startCount	*integer*	Total start reports sent.
stopCount	*integer*	Total stop reports sent.

2.354.1.3 GroupModel

flows	*list of FlowModel* see 2.354.1.3.1	List of flows.

2.354.1.3.1 FlowModel

age	*integer*	Time in seconds since flow detection.
dropReason	*string*	Reason for drop.
dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.
emtStatus	*string*	Exact match entry programming status.
ingressIntf	*string*	Flow ingress interface.

[ipProtocol]	*enum: ipProtoExperiment1 | ipProtoRsvp | ipProtoL2tp | ipProtoIpv6NoNxt | ipProtoIpip6 | ipProtoIcmp | ipProtoAh | ipProtoPim | ipProtoIpv6HopByHop | ipProtoTcp | ipProtoUdp | ipProtoIpip | ipProtoIgmp | ipProtoGre | ipProtoIpv6Route | ipProtoSctp | ipProtoIcmpv6 | ipProtoIsis | ipProtoOspf | ipProtoUnknown | ipProtoIpv6Frag | ipProtoIpv6Opts | ipProtoExperiment2 | ipProtoEsp | ipProtoVrrp*	IP protocol.
ipProtocolNumber	*integer*	IP protocol number.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.

2.355 show flow tracking mirror-on-drop counters drop-reason [...]

see 2.354

2.356 show flow tracking mirror-on-drop flow-table [...]

see 2.354

2.357 show flow tracking sampled counters [...]

[activeFlows] **	*integer*	Cumulative active flow count. **Since: 4.23.1
[clearTime] **	*floating point number*	UTC time when total counters were last cleared. **Since: 4.23.1
[expiredFlows] **	*integer*	Cumulative expired flow count. **Since: 4.23.1
[flows]	*integer*	Cumulative flow count.
ftrType **	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type. **Since: 4.24.1
[hardwareOffload] **	*HardwareOffloadCounters* see 2.357.1	Hardware flow group counters. **Since: 4.25.2
[packets] **	*integer*	Cumulative count of sampled packets. **Since: 4.23.1
running	*boolean*	Flow tracking agent is running.
[sampleCounters] **	*SampleCounters* see 2.357.2	Sample counters. **Since: 4.25.2
trackers	*table(string) = TrackerCounters* see 2.357.3	A mapping of tracker name to counters.

2.357.1 HardwareOffloadCounters

[clearTime]	*floating point number*	UTC time when counters were last cleared.
[flowsActive]	*integer*	Number of active flows in hardware.
[flowsCreated]	*integer*	Number of flows created in hardware.
[flowsDeleted]	*integer*	Number of flows deleted from hardware.
[flowsPending]	*integer*	Number of flows pending to be created in hardware.
[pktsDiscarded]	*integer*	Number of packets discarded in CPU for hardware flows.
[pktsHardwareFailed]	*integer*	Number of packets that failed to create hardware flow.
[pktsHardwareMiss]	*integer*	Number of packets that didn't hit hardware flows.
[pktsReceived]	*integer*	Number of packets received for hardware flows.

2.357.2 SampleCounters

[discarded]	*integer*	Number of samples discared.
[hardwareReceived]	*integer*	Number of hardware samples.
[poolSize]	*integer*	Sample pool size.
received	*integer*	Number of samples received.

2.357.3 TrackerCounters

activeFlows	*integer*	Number of active flows.
[clearTime]	*floating point number*	UTC time when counters were last cleared.
expiredFlows	*integer*	Cumulative expired flow count.
exporters	*table(string) = ExporterCounters* see 2.357.3.1	A mapping of exporter name to counters.
flowGroups	*table(string) = FlowGroupCounters* see 2.357.3.2	A mapping of flow group name to counters.
flows	*integer*	Cumulative flow count.
packets	*integer*	Cumulative count of sampled packets.

2.357.3.1 ExporterCounters

[clearTime] **	*floating point number*	UTC time when counters were last cleared. **Since: 4.22.1
collectors	*list of CollectorCounters* see 2.357.3.1.1	A list of per-collector counters.
exporterType	*ipfix*	Exporter type.

2.357.3.1.1 CollectorCounters

addrAndPort	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Collector IP address and port number.
exportedFlowRecordTotal-Count	*integer*	The total number of flow records sent to the collector.
exportedMessageTotalCount	*integer*	The total number of messages sent to the collector.
exportedOctetTotalCount	*integer*	The total number of bytes sent to the collector.
lastUpdates	*CollectorTimestamp* see 2.357.3.1.1.3	UTC time of last events.
sets	*table(numeric string) = CollectorSetCounters* see 2.357.3.1.1.1	A mapping of IPFIX template ID (set ID) to per-set-id counters.
templates	*list of CollectorTemplateCounters* see 2.357.3.1.1.2	A list of per-template counters.

2.357.3.1.1.1 CollectorSetCounters

flowRecords	*integer*	Number of flow records sent to the collector.

2.357.3.1.1.2 CollectorTemplateCounters

templateType	*enum: template | optionsTemplate*	IPFIX template type.

templates	*integer*	Number of templates sent to the collector.

2.357.3.1.1.3 CollectorTimestamp

[dataRecord]	*floating point number*	UTC time of last exported data record.
[message]	*floating point number*	UTC time of last exported message.
[optionsData]	*floating point number*	UTC time of last exported options data record.
[template]	*floating point number*	UTC time of last exported template.

2.357.3.2 FlowGroupCounters

activeFlows	*integer*	Number of active flows.
expiredFlows	*integer*	Cumulative expired flow count.
flows	*integer*	Cumulative flow count.
[hardwareOffload] **	*HardwareOffloadCounters* see 2.357.1	Hardware flow group counters. **Since: 4.25.2
packets	*integer*	Cumulative count of sampled packets.

2.358 show flow tracking sampled counters debug

revision 2

ftrType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type.
[packetErrorCounters]	*PacketErrorCounter* see 2.358.1	Packet error counters debug.
running	*boolean*	Flow tracking agent is running.

2.358.1 PacketErrorCounter

flowTableLimit	*integer*	Number of packets dropped due to flow table limit.
ftInputQueueFull	*integer*	Number of packets dropped due to flow table manager input queue full.
fttSimReqQueueFull	*integer*	Number of packets dropped due to simulation request queue full.
fttSimRespQueueFull	*integer*	Number of packets dropped due to simulation response queue full.
routeInfoQueueFull	*integer*	Number of times route information queue is full.

2.359 show flow tracking sampled counters interface errors [...]

`[egressCounters]`	*table(Interface name) = PerInterfaceFlowErrCounters* see 2.359.1	Egress counters per interface.
`ftrType`	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type.
`[ingressCounters]`	*table(Interface name) = PerInterfaceFlowErrCounters* see 2.359.1	Ingress counters per interface.
`running`	*boolean*	Sampled flow tracking is active.

2.359.1 PerInterfaceFlowErrCounters

`other`	*integer*	Packets dropped for other reasons.
`parseFail`	*integer*	Packets that could not be parsed.
`unknownProtocol`	*integer*	Packets with unsuppored protocol type.

2.360 show flow tracking sampled flow-table [...]

Could also be an instance of:
IntTrackingModel 2.360.1.1.1.6

ftrType **	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type. **Since: 4.23.1
running	*boolean*	Sampled flow tracking is active.
trackers	*table(string) = TrackerModel* see 2.360.1	A mapping between tracker name and tracker.

2.360.1 TrackerModel

groups	*table(string) = GroupModel* see 2.360.1.1	A mapping bewteen group name and group.
numFlows	*integer*	Total number of flows.

2.360.1.1 GroupModel

flows	*list of FlowModel* see 2.360.1.1.1	List of flows.

2.360.1.1.1 FlowModel

bytesReceived	*integer*	Number of bytes received.
[flowDetail]	*FlowDetailModel* see 2.360.1.1.1.1	Flow detailed information.
key	*FlowKeyModel* see 2.360.1.1.1.6	Flow key.
pktsReceived	*integer*	Number of packets received.
startTime	*floating point number*	Flow start time.

2.360.1.1.1.1 FlowDetailModel

Could also be an instance of:
IntFlowDetailModel 2.360.1.1.1.3

bgpNextHopIp	*IP address*	BGP next hop IP address.
[dot1qCustomerVlanId] **	*integer*	IEEE 802.1Q inner tag VLAN ID. **Since: 4.27.0
[dot1qVlanId] **	*integer*	IEEE 802.1Q tag VLAN ID. **Since: 4.27.0
dstAs	*integer*	BGP destination AS.
dstEthAddr	*MAC address*	Flow destination MAC address.
dstPrefixLen	*integer*	Destination prefix length.
egressIntf	*string*	Flow egress interface.
egressVlanId	*integer*	Flow egress VLAN ID.
[flowLabel] **	*integer*	IPv6 flow label. **Since: 4.25.0

[hwBytesReceived] **	*integer*	Number of bytes received in hardware. **Since: 4.25.2
[hwPktsReceived] **	*integer*	Number of packets received in hardware. **Since: 4.25.2
ingressIntf	*interface name*	Flow ingress interface.
ingressVlanId **	*integer*	Flow ingress VLAN ID. **Since: 4.30.0
lastPktTime	*floating point number*	Last packet received time.
[mpls] **	*MplsFlowDetail* see 2.360.1.1.1.2	MPLS flow details. **Since: 4.26.2
nextHopIp	*IP address*	Next hop IP address.
[sampledBytesReceived] **	*integer*	Number of sampled bytes received. **Since: 4.25.2
[sampledPktsReceived] **	*integer*	Number of sampled packets received. **Since: 4.25.2
srcAs	*integer*	BGP source AS.
srcEthAddr	*MAC address*	Flow source MAC address.
srcPrefixLen	*integer*	Source prefix length.
tcpFlags	*string*	Flow TCP flags.
tos	*integer*	TOS in IP header.
vni **	*string*	VXLAN Network Identifier. **Since: 4.23.1

2.360.1.1.1.2 MplsFlowDetail

[fec]	*IP address prefix*	Forwarding equivalence class assignment.
labelStack	*list of integer*	Egress MPLS label stack.
labelStackTruncated	*boolean*	The actual flow had labels beyond 'labelStack'.
[topLabel]	*integer*	Ingress MPLS top label.
[topLabelType]	*enum: ldp | teMidpt | pseudowire | vpn | bgp*	Control protocol that allocated 'topLabel'.

2.360.1.1.1.3 IntFlowDetailModel

bgpNextHopIp	*IP address*	BGP next hop IP address.
devicesInPath	*integer*	Devices in path.
devicesInformation	*list of IntFlowDetailNodeInfo* see 2.360.1.1.1.5	List of inband telemetry device information.
[dot1qCustomerVlanId] **	*integer*	IEEE 802.1Q inner tag VLAN ID. **Since: 4.27.0
[dot1qVlanId] **	*integer*	IEEE 802.1Q tag VLAN ID. **Since: 4.27.0
dstAs	*integer*	BGP destination AS.
dstEthAddr	*MAC address*	Flow destination MAC address.
dstPrefixLen	*integer*	Destination prefix length.
egressIntf	*string*	Flow egress interface.
egressVlanId	*integer*	Flow egress VLAN ID.

flowIntervalStats	*list of IntFlowDetailIntervalStats* see 2.360.1.1.1.4	List of inband telemetry interval statistics.
flowIntervals	*integer*	Flow intervals.
[flowLabel] **	*integer*	IPv6 flow label. **Since: 4.25.0
hopCountExceeded	*boolean*	Hop count exceeded.
[hwBytesReceived] **	*integer*	Number of bytes received in hardware. **Since: 4.25.2
[hwPktsReceived] **	*integer*	Number of packets received in hardware. **Since: 4.25.2
ingressIntf	*interface name*	Flow ingress interface.
ingressVlanId **	*integer*	Flow ingress VLAN ID. **Since: 4.30.0
lastPktTime	*floating point number*	Last packet received time.
[mpls] **	*MplsFlowDetail* see 2.360.1.1.1.2	MPLS flow details. **Since: 4.26.2
nextHopIp	*IP address*	Next hop IP address.
pathPackets	*integer*	Path packets.
pathTransistions	*integer*	Path transistions.
[sampledBytesReceived] **	*integer*	Number of sampled bytes received. **Since: 4.25.2
[sampledPktsReceived] **	*integer*	Number of sampled packets received. **Since: 4.25.2
srcAs	*integer*	BGP source AS.
srcEthAddr	*MAC address*	Flow source MAC address.
srcPrefixLen	*integer*	Source prefix length.
tcpFlags	*string*	Flow TCP flags.
tos	*integer*	TOS in IP header.
vni	*string*	VXLAN Network Identifier.

2.360.1.1.1.4 IntFlowDetailIntervalStats

avgLatencies	*list of integer*	Average latency per device in path (ns).
congestions	*list of boolean*	Congestion per device in path.
maxLatencies	*list of integer*	Maximum latency per device in path (ns).
minLatencies	*list of integer*	Minimum latency per device in path (ns).
pkts	*integer*	Number of packets.
timestamp	*floating point number*	Interval start time.

2.360.1.1.1.5 IntFlowDetailNodeInfo

deviceId	*integer*	Device ID.
egressPortId	*integer*	Egress port ID.
egressQueueId	*integer*	Egress queue ID.
ingressPortId	*integer*	Ingress port.
lastPacketCongestion	*boolean*	Congestion detected in the last sampled packet.
lastPacketLatency	*integer*	Last packet latency (ns).

ttl	*integer*	Inband telemetry TTL (hop count).

2.360.1.1.1.6 FlowKeyModel

direction **	*enum: egress | ingress | ingressAndEgress*	Packet Direction. **Since: 4.29.0
dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.
[ipProtocol]	*enum: ipProtoExperiment1 | ipProtoRsvp | ipProtoL2tp | ipProtoIpv6NoNxt | ipProtoIpip6 | ipProtoIcmp | ipProtoAh | ipProtoPim | ipProtoIpv6HopByHop | ipProtoTcp | ipProtoUdp | ipProtoIpip | ipProtoIgmp | ipProtoGre | ipProtoIpv6Route | ipProtoSctp | ipProtoIcmpv6 | ipProtoIsis | ipProtoOspf | ipProtoUnknown | ipProtoIpv6Frag | ipProtoIpv6Opts | ipProtoExperiment2 | ipProtoEsp | ipProtoVrrp*	IP protocol.
ipProtocolNumber	*integer*	IP protocol number.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.
vlanId	*integer*	Flow VLAN ID.
vrfName	*string*	VRF name.
ftrType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type.
running	*boolean*	Sampled flow tracking is active.
trackers	*table(string) = TrackerModel* see 2.360.1	A mapping between tracker name and tracker.

2.361 show flow tracking sampled flow-table hardware [...]

ftrType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type.
running	*boolean*	Sampled flow tracking is active.
trackers	*table(string) = HoTrackerModel* see 2.361.1	A mapping between tracker name and tracker.

2.361.1 HoTrackerModel

groups	*table(string) = HoGroupModel* see 2.361.1.1	A mapping bewteen group name and group.
numFlows	*integer*	Total number of flows.

2.361.1.1 HoGroupModel

flows	*list of HoFlowEntryModel* see 2.361.1.1.1	List of flows.

2.361.1.1.1 HoFlowEntryModel

[egressIntf] **	*interface name*	Flow egress interface. **Since: 4.28.2
[flowDetail]	*HoFlowDetailModel* see 2.361.1.1.1.1	Hardware offload flow detailed information.
hwCreateTime	*floating point number*	Hardware offload flow status entry create time.
[ingressIntf]	*interface name*	Flow ingress interface.
key	*FlowKeyModel* see 2.360.1.1.1.6	Hardware offload flow Status key.
state	*enum: hoFlowStateUnknown | outOfResources | inProgress | inHardware*	Hardware offload flow state.

2.361.1.1.1.1 HoFlowDetailModel

[dstEthAddr]	*MAC address*	Flow destination MAC address.
hwUpdateTime	*floating point number*	Hardware offload flow status entry update time.
[srcEthAddr]	*MAC address*	Flow source MAC address.
[swCreateTime]	*floating point number*	Hardware offload flow config entry create time.

2.362 show flow tracking sampled flow-table hardware counters [...]

ftrType	*enum: hardware | sampled | inbandTelemetry | mirrorOnDrop | none*	Flow tracker type.
running	*boolean*	Sampled flow tracking is active.
trackers	*table(string) = HoTrackerCounterModel* see 2.362.1	A mapping between tracker name and tracker.

2.362.1 HoTrackerCounterModel

groups	*table(string) = HoGroupCounterModel* see 2.362.1.1	A mapping bewteen group name and group.
numFlows	*integer*	Total number of flows.

2.362.1.1 HoGroupCounterModel

flows	*list of HoFlowCounterEntryModel* see 2.362.1.1.1	List of flows.

2.362.1.1.1 HoFlowCounterEntryModel

bytesReceived	*integer*	Number of bytes received.
createTime	*floating point number*	Hardware offload flow counter entry create time.
[egressIntf] **	*interface name*	Flow egress interface. **Since: 4.28.2
[ingressIntf]	*interface name*	Flow ingress interface.
key	*FlowKeyModel* see 2.360.1.1.1.6	Hardware offload flow counter key.
pktsReceived	*integer*	Number of packets received.
updateTime	*floating point number*	Hardware offload flow counter entry update time.

2.363 show flow tracking telemetry inband counters [...]

see 2.357

2.364 show flow tracking telemetry inband flow-table [...]

see 2.360.1.1.1.6

2.365 show flow-spec (ipv4 | ipv6) summary [...]

allVrfInstalledCount	*integer*	Number of flow-spec rules installed in hardware across all VRFs.
allVrfTotalCount	*integer*	Total number of flow-spec rules across all VRFs.
vrfs	*table(string) = VrfFlowspecRuleSummary* see 2.365.1	Flow-spec rule summary indexed by VRF name.

2.365.1 VrfFlowspecRuleSummary

installedCount	*integer*	Number of flow-spec rules installed in hardware.
totalCount	*integer*	Total number of flow-spec rules.
unInstalledCount	*integer*	Number of flow-spec rules not installed in hardware.

2.366 show flow-spec ipv4 [...]

vrfs	*table(string) = VrfFlowspecRule* see 2.366.1	Flow-spec rules indexed by VRF name.

2.366.1 VrfFlowspecRule

activeIntfs	*list of Interface name*	List of interfaces applied with flow-spec.
configuredIntfs **	*list of Interface name*	List of interfaces configured with flow-spec. **Since: 4.22.0
feasibleRules	*list of FeasibleFlowspecRule* see 2.366.1.1	Feasible flow-spec rules indexed by rule string.
infeasibleRules	*list of InfeasibleFlowspecRule* see 2.366.1.2	Infeasible flow-spec rules indexed by rule string.

2.366.1.1 FeasibleFlowspecRule

actions	*Actions* see 2.366.1.1.1	Actions for the rule.
matches	*Matches* see 2.366.1.1.3	Matches for the rule.
ruleId	*integer*	Flow-spec rule identifier.
ruleString	*string*	Flow-spec rule string representation.
status	*Status* see 2.366.1.1.2	Status for the rule.
[unsupportedActions] **	*Actions* see 2.366.1.1.1	Unsupported actions. **Since: **currentRelease**

2.366.1.1.1 Actions

[trafficAction] **	*TrafficAction* see 2.366.1.1.1.1	Traffic action to sample or stop traffic filtering. **Since: **currentRelease**
[trafficDrop]	*boolean*	Drop action.
[trafficMarking] **	*integer*	Mark DSCP action. **Since: 4.23.1
[trafficRate] **	*floating point number*	Police traffic based on bits per second (bps). **Since: 4.22.0
[trafficRatePkt] **	*floating point number*	Police traffic based on packets per second (pps). **Since: **currentRelease**
[trafficRedirect]	*TrafficRedirect* see 2.366.1.1.1.2	Traffic redirect action.

2.366.1.1.1.1 TrafficAction

nonTerminal	*boolean*	Traffic-filtering engine may evaluate or stops filtering based on this bit.

sample	*boolean*	Traffic sampling and logging is enabled.

2.366.1.1.1.2 TrafficRedirect

mirror	*boolean*	Mirror the packet instead of redirecting it.
[nexthop]	*IP address*	IP address of nexthop for redirect.
viaNexthop	*list of ViaNexthop* see 2.366.1.1.1.3	Redirect via the nexthop.
vrf	*string*	VRF name.

2.366.1.1.1.3 ViaNexthop

[mplsLabel] **	*integer*	MPLS label. **Since: 4.22.0
nexthopIp	*IP address*	IP address of nexthop for redirect.
[tunnelIndex] **	*integer*	Tunnel index within tunnel type. **Since: 4.22.0
[tunnelType] **	*enum: RSVP | VXLAN VTEP Encap | Nexthop | IS-IS SR | EosSdk sBFD | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | RSVP FRR | RSVP LER | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type. **Since: 4.22.0

2.366.1.1.2 Status

[byteCount] **	*integer*	Number of bytes hitting the rule. **Since: **currentRelease**
installed	*boolean*	Rule installed in hardware.
[packetCount]	*integer*	Number of packets hitting the rule.
[policeCount] **	*PoliceCount* see 2.366.1.1.2.1	Descriptive count for police. **Since: 4.29.2
[reason]	*string*	Reason for not programmed in hardware.

2.366.1.1.2.1 PoliceCount

[aboveCount]	*CounterInfo* see 2.366.1.1.2.2	Number of packets above police rate.
[belowCount]	*CounterInfo* see 2.366.1.1.2.2	Number of packets below police rate.

2.366.1.1.2.2 CounterInfo

[byteCount]	*integer*	Number of bytes.
[packetCount]	*integer*	Number of packets.

2.366.1.1.3 Matches

[destPort]	*list of NumericalRange* see 2.366.1.1.3.2	Destination port.
[destPrefix]	*IP address prefix*	Destination prefix.
[dscp]	*list of NumericalRange* see 2.366.1.1.3.2	DSCP.
[flowLabels]	*list of NumericalRange* see 2.366.1.1.3.2	Flow label.
[fragmentFlags]	*list of BitMask* see 2.366.1.1.3.1	Fragment flags.
[icmpCode]	*list of NumericalRange* see 2.366.1.1.3.2	ICMP code.
[icmpType]	*list of NumericalRange* see 2.366.1.1.3.2	ICMP type.
[nextProtocol]	*list of NumericalRange* see 2.366.1.1.3.2	Next protocol.
[packetLength]	*list of NumericalRange* see 2.366.1.1.3.2	Packet length.
[port]	*list of NumericalRange* see 2.366.1.1.3.2	Port.
[srcPort]	*list of NumericalRange* see 2.366.1.1.3.2	Source port.
[srcPrefix]	*IP address prefix*	Source prefix.
[tcpFlags]	*list of BitMask* see 2.366.1.1.3.1	TCP flags.

2.366.1.1.3.1 BitMask

mask	*integer*	Mask.
value	*integer*	Value.

2.366.1.1.3.2 NumericalRange

maxVal	*integer*	Maximum value.
minVal	*integer*	Minimum value.

2.366.1.2 InfeasibleFlowspecRule

infeasibleMatch	*string*	Infeasible match.
ruleId	*string*	Flow-spec rule identier.
ruleString	*string*	Flow-spec rule string representation.

2.367 show flow-spec ipv6 [...]

see 2.366

2.368 show flowcontrol [...]

`interfaceFlowControls`	*table(Interface name) = FlowControlData* see 2.368.1	Mapping between an interface and it's flow control data.
`[syslogInterval]` **	*floating point number*	Syslog interval for flow control pasuse frames in seconds. **Since: 4.26.2

2.368.1 FlowControlData

`rxAdminState`	*enum: desired | unknown | on | off | unsupported*	Receive Flowcontrol admin state.
`rxOperState`	*enum: unknown | on | disagree | off | unsupported*	Receive Flowcontrol operational state.
`rxPause`	*integer*	Pause packets received.
`txAdminState`	*enum: desired | unknown | on | off | unsupported*	Send Flowcontrol admin state.
`txOperState`	*enum: unknown | on | disagree | off | unsupported*	Send Flowcontrol operational state.
`txPause`	*integer*	Pause packets sent.

2.369 show forwarding destination [...]

revision 2

`details` **	*table(string) = PacketDestinationDetailModel* see 2.369.2	Platform specific details on forwarding destination. **Since: 4.28.1
`egressInterfaces`	*list of EgressInterface* see 2.369.1	List of egress interfaces.
`status`	*enum: success | timeout*	Status of the request.

2.369.1 EgressInterface

`logical`	*interface name*	Logical egress interface.
`physical`	*interface name*	Physical egress interface.

2.369.2 PacketDestinationDetailModel

`[fields]`	*table(string) = string*	A mapping of detail field name to its value.

2.370 show forwarding destination ingress-interface <ingressIntf> raw <rawPacket> [...]

revision 2

see 2.369

2.371 show fpga applications [...]

`applications`	*table(string) = FpgaApplicationInstances* see 2.56	A mapping of FPGA application name to information specific to each application.

2.372 show fpga applications compatibility

`applications`	*table(string) = FpgaApplicationsCompatibility* see 2.372.1	A mapping of FPGA application name to what is compatible with this verison of EOS.

2.372.1 FpgaApplicationsCompatibility

`maxHardwareApi`	*HardwareApiInfo* see 2.372.1.1	Maximum hardware API version.
`minHardwareApi`	*HardwareApiInfo* see 2.372.1.1	Minimum hardware API version.

2.372.1.1 HardwareApiInfo

`major`	*integer*	Hardware API major revision number.
`[minor]`	*integer*	Hardware API minor revision number.

2.373 show fpga applications profile [...]

`applications`	*table(string) = FpgaApplicationProfiles* see 2.57	A mapping of FPGA application name to information specific to each application.

2.374 show fpga switch ((mac address-table) | mac-address-table) multicast [...]

`[groups]`	*list of EthGroup* see 2.374.1	A list of EthGroups.

2.374.1 EthGroup

`igmpFlooding`	*boolean*	Overflood IGMP packets in hardware.
`igmpToCpu`	*boolean*	Copy IGMP packets to CPU.

interfaces	*list of Interface name*	Interfaces of the EthGroup.
macAddress	*MAC address*	EthGroup MAC address.
vlanId	*integer*	VLAN identifier.

2.375 show fpga switch [...]

switchCores	*table(numeric string) = FpgaSwitchCore* see 2.375.1	A mapping of FPGA Switch core ID to information specific to the FPGA Switch core instance.

2.375.1 FpgaSwitchCore

application	*ApplicationInfo* see 2.375.1.1	FPGA Application information.
featureProfile	*string*	Feature Profile Name.
hardwareApi **	*HardwareApiInfo* see 2.375.1.2	Hardware API version. **Since: 4.28.0
numPipelines **	*integer*	Number of pipelines. **Since: 4.28.0
portProfile	*string*	Port Profile Name.

2.375.1.1 ApplicationInfo

instance	*string*	FPGA Application Instance.
name	*string*	FPGA Application Name.

2.375.1.2 HardwareApiInfo

major	*integer*	Hardware API major revision number.
minor	*integer*	Hardware API minor revision number.

2.376 show fpga switch counters [...]

counters	*table(string) = MakoDebugCounterValue* see 2.376.1	Mapping between counter name and counter info.

2.376.1 MakoDebugCounterValue

firstSeen	*floating point number*	Time when counter was first incremented.
lastSeen	*floating point number*	Time when counter was most recently incremented.
value	*integer*	Value of counter.

2.377 show fpga switch ip route [...]

switchCores	*table(numeric string) = FpgaSwitchCoreRoutes* see 2.377.1	A mapping of FPGA Switch coreID to its programmed routes.

2.377.1 FpgaSwitchCoreRoutes

routes	*list of Route* see 2.377.1.1	List of programmed routes.

2.377.1.1 Route

action	*enum: ROUTE | DROP | FLOOD | CPU_LOW | CPU_HIGH*	Action to perform.
consistent	*boolean*	Whether the route and associated rewrite entries are all consistently programmed across the FPGA core.
destIntfs	*table(Interface name) = DestinationInterface* see 2.377.1.1.1	Destination interfaces.
[lpmEntries]	*list of string*	Hardware LPM entries that contribute to this route.
numEntries	*integer*	Number of hardware LPM entries that contribute to this route.
prefix	*IP address prefix*	Logical IPv4 destination subnet.
vrfId	*integer*	VRF ID.
vrfName **	*string*	VRF name. **Since: 4.29.0

2.377.1.1.1 DestinationInterface

ethAddr	*MAC address*	Rewrite MAC address.
vlanId	*integer*	Rewrite VLAN ID.

2.378 show fpga switch mapping [...]

switchCores	*table(numeric string) = PortMappings* see 2.378.1	A mapping of FPGA Switch core ID to its switch port mapping.

2.378.1 PortMappings

cpuInterfaces	*table(Interface name) = CpuPortMapping* see 2.378.1.1	The mapping of CPU ports to their internal ID.
interfaces	*table(Interface name) = PortMapping* see 2.378.1.2	The mapping of switch ports to their internal IDs.

2.378.1.1 CpuPortMapping

portId	*integer*	Internal port ID.

2.378.1.2 PortMapping

egressId	*integer*	Egress processor ID.
pipeId	*integer*	Ingress pipeline ID.
pipePortId	*integer*	Port ID within the pipeline.
portId	*integer*	Internal port ID.

2.379 show fpga switch multicast ip route [...]

switchCores	*table(numeric string) = FpgaSwitchMulticastCoreRoutes* see 2.379.1	A mapping of FPGA Switch coreID to its programmed multicast routes.

2.379.1 FpgaSwitchMulticastCoreRoutes

routes	*list of MulticastRoute* see 2.379.1.1	Programmed multicast routes.

2.379.1.1 MulticastRoute

consistent	*boolean*	Whether the route and associated rewrite entries are all consistently programmed across the FPGA core.
dmac	*MAC address*	Destination MAC address of the route.
group	*IP address*	Logical IP multicast group address.
ingressVlanId	*integer*	Ingress VLAN ID of the route.
interfaces	*table(Interface name) = MulticastDestination* see 2.379.1.1.1	A mapping of destination interface to its egress VLANs.
source	*IP address*	Logical IP multicast source address.
vlanGroupId	*integer*	VLAN group ID of the route.

2.379.1.1.1 MulticastDestination

egressVlanIds	*list of integer*	Egress VLAN IDs used for the interface.

2.380 show fpga switch parity event [...]

[events]	*list of ParityEvent* see 2.380.1	A list of parity events.
parityEventCount	*integer*	Number of parity events.

2.380.1 ParityEvent

address	*integer*	Address affected by the event.
count	*integer*	Number of times an event has occurred at this address.
impact	*enum: temporary* | *none* | *critical* | *unknown*	The impact of the event.
lastReportedTime	*floating point number*	Last time an event was reported in at this address in UTC.
sourceId	*integer*	ID of the event source.
sourceName	*string*	Name of the event source.
sourceType	*enum: register* | *other* | *unknown* | *table*	The type of the event source.
switchId	*integer*	ID of the switch instance where event occurred.

2.381 show fpga switch shared resources mbb [...]

[cmdErr]	*string*	Error Message.
[switchCores]	*table(numeric string) = MbbTables* see 2.381.1	A mapping of FPGA switch core ID to MBB tables.

2.381.1 MbbTables

agentPid	*integer*	PID of the agent handling request.
mbbTables	*table(string) = MbbTable* see 2.381.1.1	Map of MBB tables by table name.

2.381.1.1 MbbTable

[entries]	*table(numeric string) = MbbTableEntry* see 2.381.1.1.1	Map of MBB table entry by entry key.
[summary]	*MbbTableSummary* see 2.381.1.1.2	Counts of MBB table entries.

2.381.1.1.1 MbbTableEntry

[entry]	*string*	Entry.
ref	*integer*	Reference to dependant MBB table entry.
[retry]	*boolean*	Entry on retry queue.
[state]	*enum: inHw | deleted | errRetryTimeout | inProgress | errNoSpace*	Entry state.

2.381.1.1.2 MbbTableSummary

numMbbTableEntries	*integer*	Number of table entries.
numMbbTableEntriesDeleted	*integer*	Number of entries in hardware, but marked for deletion.
numMbbTableEntriesErrNoSpace	*integer*	Number of entries that will not fit in hardware.
numMbbTableEntriesErrRetryTimeout	*integer*	Number of entries that timed out waiting for dependency resolution.
numMbbTableEntriesInHw	*integer*	Number of entries in hardware.
numMbbTableEntriesInProgress	*integer*	Number of entries being programmed in hardware.
[numMbbTableEntriesOnRetryQueue]	*integer*	Number of entries on the retry queue.

2.382 show fpga switch shared resources operational status (nexthop | acl) [...]

[cmdErr]	*string*	Error Message.
[switchCores]	*table(numeric string) = OperationalStatus* see 2.382.1	A mapping of FPGA switch core ID to shared resource operational status.

2.382.1 OperationalStatus

[acl]	*Acl* see 2.382.1.1	ACL operational status.
agentPid	*integer*	PID of the agent handling request.
[nat]	*Nat* see 2.382.1.2	NAT operational status.
[nexthop]	*Nat* see 2.382.1.2	Nexthop operational status.

2.382.1.1 Acl

2.382.1.2 Nat

[natSessions]	*table(numeric string) = NatSession* see 2.382.1.2.1	Map of NAT sessions by entry key.
[natSummary]	*NatSummary* see 2.382.1.2.2	NAT operational status summary.

2.382.1.2.1 NatSession

forwardEntry	*NatEntry* see 2.382.1.2.1.1	NAT session forwarding entry.
[reverseEntry]	*NatEntry* see 2.382.1.2.1.1	NAT sesssion reverse entry.
status	*enum: inprogress | created | statusnone | teardown*	NAT session programming status.
title	*string*	Short title of the NAT session.

2.382.1.2.1.1 NatEntry

entryStatus	*NatEntryStatus* see 2.382.1.2.1.2	Active status of the NAT entry.
matchConfig	*NatMatchConfig* see 2.382.1.2.1.18	NAT match configuration from MakoNat.
rewriteConfig	*NatRewriteConfig* see 2.382.1.2.1.20	NAT rewrite configuration from MakoNat.

2.382.1.2.1.2 NatEntryStatus

direction	*enum: reverse | forward*	NAT entry direction.
l3LagIndexLocks	*list of Interface name*	Set of L3 LAG index allocations locked by this entry.

matchStatus	*NatMatchStatus* see 2.382.1.2.1.3	NAT match status.
rewriteStatus	*NatRewriteStatus* see 2.382.1.2.1.15	NAT rewrite status.
translationType	*enum: destination | source*	NAT entry translation type.

2.382.1.2.1.3 NatMatchStatus

matchL2Intfs	*NatL2Intfs* see 2.382.1.2.1.5	State of L2 interface to L3 LAG index resolution.
matchPipelineEntrySets	*table(numeric string) = NatMatchPipelineEntrySet* see 2.382.1.2.1.6	Map of pipelines to the sets of NAT match entries programmed.
matchPortMask	*integer*	Bit mask of ingress port ids configured with the NAT match.
matchPortSet	*list of ChipPortId* see 2.382.1.2.1.4	Set of chip ports with the NAT match applied.
matchVlanId	*integer*	VLAN id of the ingress interfaces.
[matchVlanStatus]	*NatMatchVlanStatus* see 2.382.1.2.1.11	NAT match VLAN status.

2.382.1.2.1.4 ChipPortId

chipPortId	*integer*	Chip port index.
interface	*interface name*	Chip port interface name.

2.382.1.2.1.5 NatL2Intfs

resolvedIntf	*table(Interface name) = integer*	Map interfaces resolved to their L3 LAG indexes.
unresolvedIntf	*list of Interface name*	Set of interfaces not yet resolved.

2.382.1.2.1.6 NatMatchPipelineEntrySet

matchPipelineEntries	*list of NatMatchPipelineEntry* see 2.382.1.2.1.7	Set of NAT match pipeline entries.

2.382.1.2.1.7 NatMatchPipelineEntry

matchKey	*NatMatchKey* see 2.382.1.2.1.8	NAT match entry key.
[matchResult]	*NatMatchResult* see 2.382.1.2.1.9	NAT match entry result.

2.382.1.2.1.8 NatMatchKey

[matchDstIp]	*IPv4 address*	Destination IP address to match.
[matchDstPort]	*integer*	Destination port to match.

[matchL3LagIndex]	*integer*	L3 LAG index to match.
[matchNatId]	*integer*	NAT id to match.
matchNatType	*enum: dnat | snapt | dyndnapt | dynsnapt | dyntwicenapt | dnapt | snat*	Type of NAT match.
[matchSrcIp]	*IPv4 address*	Source IP to match.
[matchSrcPort]	*integer*	Source port to match.

2.382.1.2.1.9 NatMatchResult

[l3LagIndex]	*integer*	NAT match entry nexthop L3 LAG index.
[natRewriteId]	*integer*	NAT match entry rewrite id.
[protocol]	*enum: all | tcp | none | udp*	NAT match entry protocol.
[rewriteIndex]	*RewriteIndex* see 2.382.1.2.1.10	NAT match entry rewrite index.
[trap]	*boolean*	A NAT match will trap packet to CPU.
[useAcl]	*boolean*	The NAT match entry uses ACL.
valid	*boolean*	The NAT match entry valid.

2.382.1.2.1.10 RewriteIndex

rewriteType	*enum: l3flood | arp*	Type of rewrite index.
rewriteValue	*integer*	Rewrite index value.

2.382.1.2.1.11 NatMatchVlanStatus

permitMask	*integer*	Bit mask of ingress port ids permitted for VLAN.
pipelineVlanMembership	*table(numeric string) = NatMatchPipelineVlanMembership* see 2.382.1.2.1.12	Map of pipelines to VLAN membership status.

2.382.1.2.1.12 NatMatchPipelineVlanMembership

[acl]	*VlanMembershipAcl* see 2.382.1.2.1.13	NAT match VLAN ACL membership.
[nat]	*VlanMembershipNat* see 2.382.1.2.1.14	NAT match VLAN NAT membership.

2.382.1.2.1.13 VlanMembershipAcl

[aclDefaultPermit]	*boolean*	Default action of ACL is permit.
[aclEnabled]	*boolean*	ACL enabled on this VLAN.
[aclId]	*integer*	ACL id.

2.382.1.2.1.14 VlanMembershipNat

[natEnabled]	*boolean*	NAT match enabled on this VLAN.
[natId]	*integer*	NAT id.

2.382.1.2.1.15 NatRewriteStatus

rewriteL2Intfs	*NatL2Intfs* see 2.382.1.2.1.5	State of L2 interface to L3 LAG index resolution.
rewritePortMask	*integer*	Bit mask of egress port ids configured with the NAT rewrite.
rewritePortSet	*list of ChipPortId* see 2.382.1.2.1.4	Ports with NAT rewrite applied.
rewriteVlanId	*integer*	VLAN id of the egress interfaces.
rewrites	*table(numeric string) = NatRewritesByPipeline* see 2.382.1.2.1.16	Map of base chip port is to set of NAT rewrites.

2.382.1.2.1.16 NatRewritesByPipeline

rewritesPerBasePort	*table(numeric string) = NatRewriteResult* see 2.382.1.2.1.17	Map of rewrite ids programmed on the egress base port pair to NAT rewrite result.

2.382.1.2.1.17 NatRewriteResult

[ip]	*IPv4 address*	New IP address to write to packet.
l3ChecksumDelta	*integer*	Correction value to validate the checksum after rewriting the packet.
[port]	*integer*	New port to write to packet.

2.382.1.2.1.18 NatMatchConfig

matchData	*NatMatchData* see 2.382.1.2.1.19	NAT match entry data fields.
matchIntf	*interface name*	Ingress interface to apply NAT match config.
matchKey	*NatMatchKey* see 2.382.1.2.1.8	NAT match entry key fields.
matchVlanId	*integer*	VLAN id of the ingress interface.

2.382.1.2.1.19 NatMatchData

[multicast]	*boolean*	NAT match is for a multicast session.
[nexthopId]	*integer*	NAT match nexthop key.
[protocol]	*enum: all | tcp | none | udp*	NAT match protocol.
[trap]	*boolean*	A NAT match will trap packet to CPU.
[useAcl]	*boolean*	The NAT match entry uses ACL.
valid	*boolean*	The NAT match entry valid.

2.382.1.2.1.20 NatRewriteConfig

rewriteData	*NatRewriteData* see 2.382.1.2.1.21	NAT rewrite data fields.
rewriteIntf	*interface name*	Egress interface to apply the NAT rewrite config.
rewriteVlanId	*integer*	VLAN id of the egress interface.

2.382.1.2.1.21 NatRewriteData

ip	*IPv4 address*	New IP address to write to packet.
l3ChecksumDelta	*integer*	Correction value to validate the checksum after rewriting the packet.
[port]	*integer*	New port to write to packet.

2.382.1.2.2 NatSummary

freeNatIds	*table(numeric string) = integer*	Number of free NAT ids per pipeline.
freeRewriteIds	*integer*	Number of free rewrite ids.
ingressPipelines	*list of integer*	Set of ingress pipelines used NAT.

2.383 show fpga switch shared resources operational status nat [...]

see 2.382

2.384 show gre tunnel static [...]

greTunnels	*table(numeric string) = L3IntfTunnelTableEntry* see 2.384.1	Collection of GRE tunnels.

2.384.1 L3IntfTunnelTableEntry

dstAddr	*IP address*	Location in the IP network where the tunnel terminates.
index	*integer*	Tunnel index.
name	*string*	Tunnel name.
srcAddr	*IP address*	Location in the IP network where the tunnel starts.
vias	*list of IpVia* see 2.151.1.2	List of tunnel vias.

2.385 show hardware abstraction table NAMESPACE TABLE_NAME [...]

entriesInTableEntryLog	*integer*	Number of entries in table entry log.
lastAcknowledgedVersion	*integer*	Last acknowledged version for the software table.
tableConfigs	*table(numeric string) = TableConfigModel* see 2.385.1	Collection of software tables keyed by unit ID.
tableName	*string*	Name of the software table.
tableNamespace	*string*	Namespace of the software table.
valid	*boolean*	True if namespace and name of the software table is valid.

2.385.1 TableConfigModel

swTableEntries	*list of TableEntryModel* see 2.385.1.1	List of software table entries.

2.385.1.1 TableEntryModel

isDelete	*boolean*	True if log entry marked for deletion.
logEntry	*LogEntryModel* see 2.385.1.1.1	Log entry.
rsrcKeys	*string*	Resource Keys.
version	*integer*	Log entry version.

2.385.1.1.1 LogEntryModel

2.386 show hardware abstraction table NAMESPACE TABLE_NAME summary

entriesInTableEntryLog	*integer*	Number of entries in table entry log.
lastAcknowledgedVersion	*integer*	Last acknowledged version for the software table.
tableConfigs	*table(numeric string) = TableConfigSummaryModel* see 2.386.1	Collection of software tables keyed by unit ID.
tableName	*string*	Name of the software table.
tableNamespace	*string*	Namespace of the software table.
valid	*boolean*	True if namespace and name of the software table is valid.

2.386.1 TableConfigSummaryModel

entriesInConfig	*integer*	Number of entries in table config.
entriesMarkedforDeletion	*integer*	Number of entries marked for deletion.

validEntriesInConfig	*integer*	Number of entries not marked for deletion.

2.387 show hardware asset-tag [...]

revision 2

assets	*table(string) = Asset* see 2.387.1	Assets for this system.
missing	*table(string) = Missing* see 2.387.2	Missing assets.

2.387.1 Asset

serial	*string*	Factory-set serial number.
tag	*string*	Customer-supplied asset tag.

2.387.2 Missing

fruType	*string*	Type of the missing FRU.
tag	*string*	Customer-supplied asset tag.

2.388 show hardware capacity [...]

tables	*list of TableEntry* see 2.388.1	A list of tables and utilization.

2.388.1 TableEntry

chip	*string*	Name of chip. Populated only for divergent (per-chip) features.
committed	*integer*	Block of entries dedicated to this feature, but may or may not be in use now. This value is applicable only to those tables which allocate a block of resources (like Bank) to a given feature.
feature	*string*	Name of feature using the given hardware table. When this column is empty, it is an aggregate of the usage by all features using this table.
free	*integer*	Number of free entries: 'Max entries available now - Used entries', where maximum available entries exclude entries used by and committed to other features.
highWatermark	*integer*	The maximum number of entries used by this feature so far. This value may get reset after agent restart or SSO switchover.

maxLimit	*integer*	Best case maximum entries available for this feature, considering no other feature is using them. This is the theoretical maximum resource utilization possible for this feature.
table	*string*	Name of hardware resource / table.
used	*integer*	Number of entries currently in use for the table by given feature.
usedPercent	*integer*	Used percent (%) calculated as 'Number of entries in use by the given feature / (Number of entries in use by the given feature + Number of free entries) * 100'.

2.389 show hardware capacity alert threshold

thresholds	*table(string) = TableThreshold* see 2.389.1	A mapping from table name to thresholds.

2.389.1 TableThreshold

configThreshold	*integer*	Configured Threshold in percent (%).
defaultThreshold	*integer*	Default Threshold in percent (%).

2.390 show hardware clock frequency sink [...]

frequencySinks	*table(string) = FreqSinkInfo* see 2.390.1	Frequency sinks indexed by their name.

2.390.1 FreqSinkInfo

description	*string*	Description of the component.
source	*string*	Name of the frequency source the sink is using.

2.391 show hardware clock frequency source [...]

frequencySources	*table(string) = FreqSourceInfo* see 2.391.1	Frequency sources indexed by their name.

2.391.1 FreqSourceInfo

adjustment	*floating point number*	Configured adjustment to frequency as pulse per billion.
description	*string*	Description of the component.
frequency	*integer*	Frequency in Hertz of the source.

2.392 show hardware clock frequency topology

sinks	*table(string) = SinkInfo* see 2.392.1	Sinks indexed by their name.

2.392.1 SinkInfo

availableSources	*list of string*	Available sources that the sink could use.
source	*string*	Name of the source the sink is using.

2.393 show hardware clock strobe sink [...]

strobeSinks	*table(string) = StrobeSinkInfo* see 2.393.1	Strobe sinks indexed by their name.

2.393.1 StrobeSinkInfo

description	*string*	Description of the component.
source	*string*	Name of the strobe source the sink is using.

2.394 show hardware clock strobe source [...]

strobeSources	*table(string) = StrobeSourceInfo* see 2.394.1	Strobe sources indexed by their name.

2.394.1 StrobeSourceInfo

configuredAdjustment	*floating point number*	Configured adjustment.
description	*string*	Description of the component.
frequency	*integer*	Frequency in Hertz of the strobe.
internalAdjustment	*floating point number*	Internal adjustment at the time of manufacturing.

2.395 show hardware clock strobe topology

see 2.392

2.396 show hardware clocking

ports	*table(string) = ClockPortStatus* see 2.396.1	Clock ports indexed by their name.

2.396.1 ClockPortStatus

[cableDelay]	*floating point number*	Cable delay in nanoseconds unit.
signal	*enum: none* | *gnssAntenna* | *gnssAntennaActive* | *pps*	Port signal source.
[status] **	*enum: down* | *up*	Ports signal status. **Since: 4.30.1

2.397 show hardware counter drop [...]

dropEvents	*table(string) = DropEvents* see 2.397.1	Mapping from chipName to drop events.
totalAdverseDrops	*integer*	Total count of adverse drop counters.
totalCongestionDrops	*integer*	Total count of congestion drop counters.
totalPacketProcessorDrops	*integer*	Total count of packet processor drop counters.

2.397.1 DropEvents

dropEvent	*list of DropEvent* see 2.397.1.1	List of drop events.

2.397.1.1 DropEvent

counterId	*integer*	Counter ID.
counterName	*string*	Counter name.
counterType	*string*	Counter type.
dropCount	*integer*	Drop count.
[dropInLastMinute]	*integer*	The number of drops that happend in the last minute.
[dropInLastOneDay]	*integer*	The number of drops that happend in the last one day.
[dropInLastOneHour]	*integer*	The number of drops that happend in the last one hour.
[dropInLastOneWeek]	*integer*	The number of drops that happend in the last one week.
[dropInLastTenMinute]	*integer*	The number of drops that happend in the last ten minutes.
eventCount	*integer*	Drop event count.
initialEventTime	*string*	UTC timestamp of first occurrence.
[initialThresholdEventTime]	*string*	UTC timestamp of first occurrence above configured threshold.
lastEventTime	*string*	UTC timestamp of most recent occurrence.
[lastSyslogTime]	*string*	UTC timestamp of most recent syslog.
[lastThresholdEventTime]	*string*	UTC timestamp of most recent occurrence above configured threshold.
[thresholdEventCount]	*integer*	Drop count above configured threshold count.

2.398 show hardware counter events

counterEvents	*table(numeric string) = CounterEvent* see 2.398.1	Mapping between counter Id and information for events for a counter.

2.398.1 CounterEvent

counterEventName	*string*	Name of the counter causing the event.
fiveMinuteEvents	*table(string) = ChipEvent* see 2.398.1.1	Mapping between chip name and the events over five minutes for a chip.
oneDayEvents	*table(string) = ChipEvent* see 2.398.1.1	Mapping between chip name and the events over one day for a chip.
oneHourEvents	*table(string) = ChipEvent* see 2.398.1.1	Mapping between chip name and the events over one hour for a chip.
oneWeekEvents	*table(string) = ChipEvent* see 2.398.1.1	Mapping between chip name and the events over one week for a chip.
tenMinuteEvents	*table(string) = ChipEvent* see 2.398.1.1	Mapping between chip name and the events over ten minutes for a chip.

2.398.1.1 ChipEvent

eventCount	*integer*	Number of times the event has occurred.
firstOccurrence	*floating point number*	Time of first occurrence.
lastOccurrence	*floating point number*	Time of last occurrence.
zScore	*floating point number*	Number of standard deviations away from mean the current counter event value is.

2.399 show hardware counter feature

[aclEgressIpv4]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress IPv4 ACL counters.
[aclEgressIpv6]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress IPv6 ACL counters.
[aclEgressMac] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress MAC ACL counters. **Since: 4.24.0
[aclIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress ACL counters.
[cpuTrafficPolicyIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for CPU traffic policy counters. **Since: 4.23.0
[decapGroup]	*CounterFeatureInfo* see 2.399.1	Resource and status information for decap group counters.
[directFlow] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for DirectFlow counters. **Since: 4.24.1
[ecn] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress ECN counters. **Since: 4.22.1
[flowspec]	*CounterFeatureInfo* see 2.399.1	Resource and status information for Flowspec counters.
[flowspecTwoColor] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for Flowspec two-color counters. **Since: 4.29.2
[greTunnelInterfaceEgress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress GRE tunnel interface counters. **Since: 4.22.0
[greTunnelInterfaceIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress GRE tunnel interface counters. **Since: 4.22.0
[intfPolicingIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress interface policing counters. **Since: 4.27.0
[intfPolicingMeterIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress interface policing meters. **Since: 4.27.0
[intfTrafficPolicyEgress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress interface traffic policy counters. **Since: 4.27.0
[intfTrafficPolicyIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress interface traffic policy counters. **Since: 4.23.1
[ipInIpEncap] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for IP-in-IP encap counters. **Since: 4.29.1
[ipv4v6Egress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress IP counters. **Since: 4.23.0
[ipv4v6Ingress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress IP counters.

[l2SubInterfaceEgress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress layer 2 subinterface bridged packet counters.
[l2SubInterfaceIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress layer 2 subinterface counters.
[l2SubInterfaceRouted-Egress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress layer 2 subinterface routed packet counters. **Since: 4.28.1
[l3Ipv4v6Egress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for layer 3 egress IPv4, IPv6 counters. **Since: 4.23.1
[l3Ipv4v6Ingress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for layer 3 ingress IPv4, IPv6 counters. **Since: 4.23.1
[maxQueueSize]	*CounterFeatureInfo* see 2.399.1	Resource and status information for LANZ counters.
[mplsInterfaceIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for MPLS interface ingress counters.
[mplsLfib]	*CounterFeatureInfo* see 2.399.1	Resource and status information for MPLS LFIB counters.
[mplsTunnel]	*CounterFeatureInfo* see 2.399.1	Resource and status information for MPLS tunnel counters.
[multicastIpv4]	*CounterFeatureInfo* see 2.399.1	Resource and status information for IPv4 multicast counters.
[multicastIpv6] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for IPv6 multicast counters. **Since: 4.23.2
[nexthop]	*CounterFeatureInfo* see 2.399.1	Resource and status information for nexthop group counters.
[oamLmEgress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress OAM loss measurement. **Since: 4.27.2
[oamLmIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress OAM loss measurement. **Since: 4.27.2
[oamSyntheticLm] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for OAM synthetic loss measurement. **Since: 4.28.0
[pbrIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress PBR counters. **Since: 4.23.2
[pdp]	*CounterFeatureInfo* see 2.399.1	Resource and status information for PDP counters.
[qosIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress QoS counters.
[qosMeterIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress QoS meters. **Since: 4.23.1
[qosTrTcmIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress QoS trTCM counters. **Since: 4.24.0

[queueEgress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress queue counters. **Since: 4.23.0
[realTimeTrafficBasedWfq] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for real-time traffic-based weighted fair queueing counters. **Since: 4.25.0
[rfc2544NetworkBenchmarks] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for RFC2544 counters. **Since: 4.28.1
[routeIpv4]	*CounterFeatureInfo* see 2.399.1	Resource and status information for IPv4 route counters.
[routeIpv6]	*CounterFeatureInfo* see 2.399.1	Resource and status information for IPv6 route counters.
[routedPortIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress routed port counters.
[sampledFlowTracking] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for sampled flow tracking counters. **Since: 4.24.2
[segmentSecurity] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for segment security counters. **Since: 4.29.2
[sflowAccel] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for sFlow accel counters. **Since: 4.23.2
[stormControlMeterIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress storm control metering. **Since: 4.25.1
[subInterfaceEgress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress layer 3 subinterface counters.
[subInterfaceIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress layer 3 subinterface counters.
[subInterfaceStormControl-MeterIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress storm control metering on subinterfaces. **Since: 4.30.1
[sviEgress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress SVI counters.
[sviIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress SVI counters.
[tapAggIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress Tap Aggregation counters. **Since: 4.24.2
[tcIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress traffic-class counters. **Since: 4.30.0
[vlanEgress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for egress VLAN counters.
[vlanIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress VLAN counters.
[vlanTcIngress]	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress layer 3 interface traffic-class counters.

[vniDecap]	*CounterFeatureInfo* see 2.399.1	Resource and status information for VNI decapsulation counters.
[vniEncap]	*CounterFeatureInfo* see 2.399.1	Resource and status information for VNI encapsulation counters.
[voqIngress] **	*CounterFeatureInfo* see 2.399.1	Resource and status information for ingress VOQ counters. **Since: 4.23.0
[vtepDecap]	*CounterFeatureInfo* see 2.399.1	Resource and status information for VTEP decapsulation counters.
[vtepEncap]	*CounterFeatureInfo* see 2.399.1	Resource and status information for VTEP encapsulation counters.

2.399.1 CounterFeatureInfo

[resourceTypes]	*table(string) = CounterFeatureResource* see 2.399.1.1	A mapping between a resource type and counter resources.
[status]	*enum: degraded | down | up | unknown*	Functional status of the counter feature.
[statusDetail]	*string*	Detailed description of the counter feature status.

2.399.1.1 CounterFeatureResource

resourceIds	*list of integer*	Counter resource list.

2.400 show hardware eeprom [...]

components **	*table(string) = Component* see 2.400.1	Mapping of component names to their PreFDL fields. **Since: 4.26.0

2.400.1 Component

[asy]	*string*	Assembly number.
[certificate]	*string*	ASCII readable PEM format (Base64 encoded) certificate.
[cpuFdlVariables]	*table(string) = string*	FDL variables for CPU card.
[cpuHwApi]	*string*	HwApi for CPU card.
[cpuKvn]	*string*	KVN for CPU card.
[cpuMfgTime]	*string*	MfgTime for CPU card.
[cpuPca]	*string*	PCA for CPU card.
[cpuSerialNumber]	*string*	Serial Number for CPU card.
[cpuSid] **	*string*	SID for CPU card. **Since: **currentRelease**
[cpuSku]	*string*	SKU for CPU card.
[crc]	*string*	Last four bytes of the PreFDL. This is a CRC-32 checksum.
[dataFormat]	*string*	First four bytes of the PreFDL. This self-describes the PreFDL format.
[deviation]	*list of string*	Defines an action that was taken that was not part of theoriginal build instructions.
[fdlVariables]	*table(string) = string*	Python code (a dictionary) allowing for arbitrary variables to be set.
[hwApi]	*string*	Hardware API.
[hwEpoch]	*string*	Hardware epoch.
[hwRev]	*string*	Hardware revision. This field is used to display the hardware revision to the user.
[kvn]	*string*	The KVN verification field is no longer used. Because it is a required field in formats 0000, 0001, and 0002, the default value 111 is stored as a placeholder.
[macAddrBase]	*MAC address*	The first MAC address allocated to the board.
[mfgTime]	*string*	When the PreFDL was generated.
[mfgTime2] **	*string*	The system clock when the PreFDL was generated, deprecates MfgTime. **Since: 4.28.2
[other]	*list of string*	Present fields which are uncommon.
[pca]	*string*	Printed circuit assembly number.
[serialNumber]	*string*	Board serial number.
[sid]	*string*	System ID. Used by run-time software (along with the HwAPI) to identify the system and choose a FDL.
[signature]	*string*	ASCII readable PEM format (Base64 encoded) signature.
[signatureList]	*list of string*	PreFDL values that should be used in validating the signature.

[sku]	*string*	Software-display SKU.
[ztpToken]	*string*	JWT token to validate the switch in SecureZTP.

2.401 show hardware external alarm-input

alarms	*list of ExternalAlarm* see 2.401.1	List of alarm statuses.

2.401.1 ExternalAlarm

action	*string*	Alarm action.
alarmId	*integer*	Alarm ID number.
asserted	*boolean*	Alarm asserted.
count **	*integer*	Number of times alarm has asserted. **Since: 4.21.3
description	*string*	Alarm description.
lastAssertTime	*floating point number*	Time of last assertion.
lastDeassertTime	*floating point number*	Time of last deassertion.
polarity	*string*	Alarm polarity.

2.402 show hardware external alarm-input ALARM_ID

see 2.401.1

2.403 show hardware forwarding profile

linecards	*table(string) = string*	A mapping between a linecard and its forwarding profile.
systemProfile	*string*	Configured system default forwarding profile.

2.404 show hardware fpga detail

revision 2

systemComponents	*table(string) = FpgaHardwareDetails* see 2.404.1	A mapping of system devices to its FPGAs.

2.404.1 FpgaHardwareDetails

fpgas	*table(string) = FpgaHardwareDetail* see 2.404.1.1	Detailed information for the FPGAs in the system.

2.404.1.1 FpgaHardwareDetail

`fpgaClassName`	*string*	Name of FPGA.
`isHitlessResetSupported`	*enum: False | N/A | True*	FPGA support for hitless resets.
`majorVersion`	*integer*	Major version number of FPGA.
`minorVersion`	*integer*	Minor version number of FPGA.

2.405 show hardware fpga error

Could also be an instance of:
FpgaCrcErrorsFixed 2.405.1
FpgaCrcErrorsModular 2.405.1

correctableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its correctable CRC-Error details. On modular systems the FPGA name contains the slot number.
fpgaCrcRecoveryMode	*enum: powerCycle | log | autoRepair*	System action on FPGA CRC-Error.
[linecardFpgaCrcRecovery-Mode]	*enum: powerCycle | log*	Linecard action on FPGA CRC-Error.
softwareCorrectedCrcs **	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its software-repaired CRC-Error details. On modular systems the FPGA name contains the slot number. **Since: 4.21.3
uncorrectableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its uncorrectable CRC-Error details. On modular systems the FPGA name contains the slot number.

2.405.1 FpgaCrcErrorDetails

count	*integer*	Number of CRC-Error occurrences.
[firstOccurrence]	*floating point number*	First CRC-Error occurrence.
[lastOccurrence]	*floating point number*	Last CRC-Error occurrence.
correctableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its correctable CRC-Error details. On modular systems the FPGA name contains the slot number.
fpgaCrcRecoveryMode	*enum: powerCycle | log | autoRepair*	System action on FPGA CRC-Error.
[linecardFpgaCrcRecovery-Mode]	*enum: powerCycle | log*	Linecard action on FPGA CRC-Error.
softwareCorrectedCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its software-repaired CRC-Error details. On modular systems the FPGA name contains the slot number.
uncorrectableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its uncorrectable CRC-Error details. On modular systems the FPGA name contains the slot number.
correctableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its correctable CRC-Error details. On modular systems the FPGA name contains the slot number.
fpgaCrcRecoveryMode	*enum: powerCycle | log | autoRepair*	System action on FPGA CRC-Error.
[linecardFpgaCrcRecovery-Mode]	*enum: powerCycle | log*	Linecard action on FPGA CRC-Error.
softwareCorrectedCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its software-repaired CRC-Error details. On modular systems the FPGA name contains the slot number.
uncorrectableCrcs	*table(string) = FpgaCrcErrorDetails* see 2.405.1	Maps FPGA name to its uncorrectable CRC-Error details. On modular systems the FPGA name contains the slot number.

2.406 show hardware logical-port (pool | POOL_RANGE) status

pools	*table(string) = PoolStatus* see 2.406.1	Mapping between a logical port pool's name and its status.

2.406.1 PoolStatus

configured	*integer*	Number of configured ports in the pool.
free	*integer*	Number of free logical ports in the pool.
interfaces	*list of Interface name*	Member interfaces.
maximum	*integer*	Maximum logical ports in the pool.

2.407 show hardware pll [...]

clocks	*table(string) = ClockModel* see 2.407.1	Clock name to clock information.

2.407.1 ClockModel

description	*string*	PLL description.
model	*string*	PLL manufacturer and model.
role	*enum: system | servo | timeSync | timeSyncFreeRun | phy*	PLL function.
state	*enum: freeRunning | disabled | active | initializing | failed*	Current state of the PLL.

2.408 show hardware resource [...]

tables	*table(string) = AgentName* see 2.408.1	A mapping of table name to its agents.

2.408.1 AgentName

agents	*table(string) = Unit* see 2.408.1.1	A mapping of agent name to its units.

2.408.1.1 Unit

units	*table(string) = View* see 2.408.1.1.1	A mapping of unit to its views.

2.408.1.1.1 View

views	*table(string) = Table* see 2.408.1.1.1.1	A mapping of view to its hardware table entries.

2.408.1.1.1.1 Table

entries	*list of TableEntry* see 2.408.1.1.1.2	Hardware table entries.

2.408.1.1.1.2 TableEntry

dataFields	*table(string) = string*	A mapping of entry data field attr to field val.
keyFields	*table(string) = string*	A mapping of entry key field attr to field val.

2.409 show hardware speed-group | SPEED-GROUP configurations

groups	*table(string) = SpeedGroupCap* see 2.409.1	Mapping between a speed group's name and its capabilities.

2.409.1 SpeedGroupCap

availableSpeed-Configurations	*table(string) = string*	Mapping between a SerDes speed and available speed configurations.
interfaces	*list of Interface name*	Group member interfaces.

2.410 show hardware speed-group | SPEED-GROUP status

groups	*table(string) = SpeedGroupStatus* see 2.410.1	Mapping between a speed group's name and its status.

2.410.1 SpeedGroupStatus

activeSerdesSpeeds	*list of SerdesSpeed* see 2.410.1.1	Active serdes speeds.
configuredSerdesSpeeds	*list of SerdesSpeed* see 2.410.1.1	Configured serdes speeds.
defaultSerdesSpeeds	*list of SerdesSpeed* see 2.410.1.1	Default serdes speeds.
interfaces	*list of Interface name*	Group member interfaces.
masterInterface	*interface name*	Master interface of the group.
[speedGroupAutoMode] **	*boolean*	Auto-configure speed-group setting. **Since: 4.28.0
supportedSerdesSpeeds	*list of SerdesSpeed* see 2.410.1.1	Supported serdes speeds.

2.410.1.1 SerdesSpeed

speed	*enum: 100g | 25g | 10g | 50g*	SerDes speed.

2.411 show hardware system forwarding-chips

chips	*table(string) = ForwardingChip* see 2.411.1	Forwarding chips on this system, keyed by slot and chip label.

2.411.1 ForwardingChip

chipLabel	*string*	Chip label.
chipName	*string*	Chip name.
chipVer	*string*	Chip version.
forwardingAgent	*string*	Forwarding agent.
slotLabel	*string*	Slot label.

2.412 show hardware tcam profile ((PROFILE ((feature (((acl vlan ip) | (acl vlan ipv6) | (acl vlan ipv6 egress) | (acl subintf ip) | (acl subintf ipv6) | (acl subintf mac) | (acl port ip) | (acl port ipv6) | (acl port mac) | (acl gre tunnel interface ip) | (acl port ip egress) | (acl port ip egress disabled) | (acl port ip egress mpls-tunnelled-match) | (acl port ipv6 egress) | (acl port ipv6 source-only egress) | (acl vlan ipv6 source-only egress) | (acl port mac egress) | (acl vlan ip egress) | (acl port ip ingress) | (acl vlan ip ingress) | (acl port ipv6 ingress) | (acl vlan ipv6 ingress) | (acl port mac ingress) | (acl port ipv6 source-only ingress) | (acl tunnel interface ip) | (pbr ip) | (pbr ipv6) | (pbr mpls) | (pbr subintf ip) | (pbr subintf ipv6) | (pbr tunnel interface ip) | (tunnel ipv4) | (tunnel vxlan) | (tunnel vxlan multicast mlag) | (tunnel vxlan multi-homing) | (tunnel vxlan multi-homing domain remote) | (tunnel vxlan routing) | (tunnel vxlan routing v6-underlay) | (tunnel vxlan vni policing) | (qos ip) | (qos ipv6) | (qos subintf ip) | (qos subintf ipv6) | (qos port ip) | (qos port ipv6) | (qos vlan ip) | (qos vlan ipv6) | (qos port ip egress) | (qos vlan ip egress) | (qos vlan ipv6 egress) | (qos mac) | (tapagg port) | (tapagg ip) | (tapagg ipv6) | (tapagg mac) | (tapagg tunnel termination) | (mirror ip) | (mirror ipv6) | (mirror mac) | (mirror mpls-over-gre ip) | (mirror mpls-over-gre ipv6) | (mirror port egress rate-limit) | (hash mpls-over-gre inner ipv4) | (hash mpls-over-gre inner ipv6) | (evpn mpls irb) | (evpn mpls multi-homing) | flow | (flow mac) | (flow mac vlan-rewrite) | (flow http) | (flow mac-ipv4-l4) | (flow ipv6) | (flow ipv6 l4) | (flow ipv6 vlan-tag) | cbf | (flex-route) | (counter multicast ipv4) | mpls | (mpls pop ingress) | (counter lfib) | (counter vlan-interface ingress) | (counter vlan traffic-class) | (counter vtep decap) | (flow-spec port ipv4) | (flow-spec port ipv6) | (l2-protocol forwarding) | ipsec | (ipsec egress padding-removal) | (traffic-policy cpu ipv4) | (traffic-policy cpu ipv6) | (traffic-policy port ipv4) | (traffic-policy port ipv6) | (traffic-policy port ipv4 egress) | (traffic-policy port ipv6 egress) | (traffic-policy vlan-bridged ipv4) | (traffic-policy vlan-bridged ipv6) | (traffic-policy vlan-interface ipv4) | (traffic-policy vlan-interface ipv6) | (segment-security ipv4) | (segment-security ipv6) | (forwarding-destination) | (forwarding-destination mpls) | (tcp-mss-ceiling ip) | (macsec-proxy) | (interface-policing) | (telemetry postcard policy ipv4) | (telemetry postcard policy ipv6) | (flow tracking sampled ipv4) | (ptp e2etransparent one-step) | (vlan editing extended-64) | (vrf selection) | (vrf selection extended) | cfm | rfc2544 | (mlag egress filter disabled) | (unicast rpf lookup-vrf) | (segment-routing ipv6) | (sflow filtered-sampling) | (sflow subintf) |

2.412.1 ProfileModel

[actionOverridesSystemRule] **	*list of string*	Enable action to override system rules. **Since: 4.28.0
[associatedLinecards]	*list of string*	A mapping between TCAM profile and linecards.
features	*table(string) = FeatureModel* see 2.412.1.1	List of features in current TCAM profile.
[globalFieldSizes] **	*table(string) = integer*	List of global key field size configurations. **Since: **currentRelease**

2.412.1.1 FeatureModel

actionPriority **	*integer*	Relative action priority of current feature. **Since: 4.27.1
actions	*list of string*	List of actions configured in current feature.
keyFields	*list of string*	List of key fields configured in current feature.
[keyPolicerUnsharedAction-Size] **	*integer*	Numeric policy policer unshared action size for current feature. **Since: 4.27.1
[keyPolicerUnsharedQual-Size] **	*integer*	Numeric policy policer unshared qualifier size for current feature. **Since: 4.27.1
keyPortQualSize **	*integer*	Numeric port qualifier size for current feature. **Since: 4.23.0
[keyPrefixLabelSize] **	*integer*	Numeric IP prefix lookup label size for current feature. **Since: 4.26.2
[keyPrefixLpmLabelSize] **	*integer*	Numeric IP prefix longest prefix match lookup label size for current feature. **Since: 4.26.2
keySizeLimit	*integer*	Maximum number of bits used as key.
packetTypes **	*list of string*	List of packet types in current TCAM profile. **Since: 4.20.5
sequence	*integer*	Numeric sequence of current feature.
sliceReservationStages **	*list of string*	List of TCAM slice reservation configuration per stage. **Since: 4.29.0

2.413 show hardware tcam profile PROFILE

[actionOverridesSystemRule] **	*list of string*	Enable action to override system rules. **Since: 4.28.1
[associatedLinecards]	*list of string*	List of TCAM profile associated linecards.
features	*table(string) = FeatureModel* see 2.412.1.1	A mapping of features in current TCAM profile.

profileName	*string*	Name of the profile.

2.414 show hardware-license tpm

family	*string*	The main type/version of the TPM.
specDate	*string*	The month, day, and year in which the TPM specification was published.
specLevel	*integer*	The specification level.
specRevision	*floating point number*	The specification revision.
vendorId	*string*	A vendor identifier that is unique to each TPM manufacturer.
vendorStr	*string*	The vendor id string of the TPM manufacturer.

2.415 show health-monitor counters

revision 2

hmCounters	*table(string) = HmNhgCounterEntry* see 2.415.1	A mapping of nexthop-group to server counters.

2.415.1 HmNhgCounterEntry

servers	*table(IPv4 or IPv6 address) = HmServerCounterEntry* see 2.415.1.1	A mapping of server IP to counters.

2.415.1.1 HmServerCounterEntry

probeType	*table(string) = HmProbeTypeCounterEntry* see 2.415.1.1.1	A mapping of probe-type to counters.

2.415.1.1.1 HmProbeTypeCounterEntry

numChecks	*integer*	Total number of health-checks for this server.
numConnectFailures	*integer*	Total number of health-check connect failures for this server.
numFailures	*integer*	Total number of health-check failures for this server.
numResponses	*integer*	Total number of health-check responses for this server.
numTimeouts	*integer*	Total number of health-check timeouts for this server.

2.416 show health-monitor probe delay [...]

intervals	*table(string) = HmPbEntry* see 2.416.1	A mapping of interval in seconds to probes with delay within that interval.

2.416.1 HmPbEntry

numProbes	*integer*	Number of probes in this interval.
percentProbes	*floating point number*	Percentage of probes in this interval.

2.417 show health-monitor status

revision 2

probeStatus	*table(string) = HealthMonitorProbeStatus* see 2.417.1	A mapping of nexthop-group name to its probe status.

2.417.1 HealthMonitorProbeStatus

serverStatus	*table(IPv4 or IPv6 address) = HealthMonitorStatusEntry* see 2.417.1.1	A mapping of Server IP to its health-monitor status.

2.417.1.1 HealthMonitorStatusEntry

adminState	*enum: shutdown | enabled*	Administrative state.
operState	*enum: down | up*	Operational state.
weight	*integer*	Server weight.

2.418 show health-monitor status detail

hmStatusDetail	*table(string) = HealthMonitorServerProbeStatusDetail* see 2.418.1	A mapping of nexthop-group name to its probe status.

2.418.1 HealthMonitorServerProbeStatusDetail

servers	*table(IPv4 or IPv6 address) = HealthMonitorStatusDetailEntry* see 2.418.1.1	A mapping of Server IP to health-monitor status detail.

2.418.1.1 HealthMonitorStatusDetailEntry

probeType	*table(string) = HealthMonitorStatusEntry* see 2.417.1.1	A mapping of probe-type to its health-monitor status.

2.419 show health-monitor summary

configProcessed	*boolean*	Config processing is completed.
hostConfigFile	*string*	Host config file.
lbSummary	*HealthMonitorLbSummary* see 2.419.1	LB summary.
masterConfigFile	*string*	Master config file.
monitoring	*boolean*	Monitoring is enabled.

2.419.1 HealthMonitorLbSummary

caFile	*string*	CA file configured.
probeInterval	*integer*	Probe interval in seconds.
sslProfile	*string*	SSL profile configured.
sslProfileValid	*boolean*	SSL profile is valid.

2.420 show health-monitor timestamps

revision 2

hmTimestamps	*table(string) = HmNhgTimestampEntry* see 2.420.1	A mapping of nexthop-group to server timestamps.

2.420.1 HmNhgTimestampEntry

servers	*table(IPv4 or IPv6 address) = HmServerTimestampEntry* see 2.420.1.1	A mapping of server IP to timestamps.

2.420.1.1 HmServerTimestampEntry

probeType	*table(string) = HmProbeTypeTimestampEntry* see 2.420.1.1.1	A mapping of probe-type to timestamps.

2.420.1.1.1 HmProbeTypeTimestampEntry

lastCheckTime	*integer*	Last time (UTC) this health-check was conducted.
lastResponseTime	*integer*	Time taken by the last health-check to connect in ms.
nextCheckTime	*integer*	Estimated next time (UTC) this health-check will be conducted.

2.421 show hostname

fqdn	*string*	The fully qualified domain name of the switch.
hostname	*string*	The current hostname of the switch.

2.422 show hosts

domainList	*list of string*	List of additional domains to lookup domains with.
[domainName]	*string*	The domain name that the listed hosts are associated with.
hosts	*table(string) = IpAddresses* see 2.422.2	A mapping between host names and their IP addresses.
nameServerConfigs **	*list of NameServerConfig* see 2.422.1	List of configured name servers. **Since: 4.26.0
nameServers	*list of string*	List of domain name servers used to resolve names.

2.422.1 NameServerConfig

ipAddr	*IP address*	Name server IP address.
priority	*integer*	Name server priority.
vrf	*string*	Name server VRF.

2.422.2 IpAddresses

ipv4Addresses	*list of IPv4 address*	List of IPV4 addresses.
ipv6Addresses	*list of IPv6 address*	List of IPV6 addresses.

2.423 show hsc arp sources SOURCE [...]

revision 2

arpSrcs	*table(MAC address) = HscArpSourceModel* see 2.423.1	A mapping between MAC address of HW VTEP and ARP entry.

2.423.1 HscArpSourceModel

uuid	*string*	The UUID of the ARP entry.
vtepIp	*IP address*	The VTEP IP corresponding to a MAC address.

2.424 show hsc certificate

certificate	*string*	SSL certificate used for OVSDB server to manager connections.

2.425 show hsc logical router [...]

routers	*table(string) = HscLogicalRouterModel* see 2.425.1	A mapping between logical router name and logical router.

2.425.1 HscLogicalRouterModel

staticRoutes	*table(IPv4 or IPv6 address prefix) = HscStaticRouterNHopsModel* see 2.425.1.2	A mapping between an IP prefix and next hop IPs.
switchBindings	*table(IPv4 or IPv6 address address and mask length) = HscSimpleLogicalSwitchModel* see 2.425.1.1	A mapping between IP/Subnet and logical switch.
uuid	*string*	The UUID of the logical router.

2.425.1.1 HscSimpleLogicalSwitchModel

displayName	*string*	The logical switch name.
[vni]	*integer*	The VXLAN Network Identifier (VNI).

2.425.1.2 HscStaticRouterNHopsModel

nhops	*list of IP address*	Next hop IP addresses.

2.426 show hsc logical switch [...]

revision 2

switches	*table(string) = HscLogicalSwitchModel* see 2.426.1	A mapping between uuid and configured logical switch.

2.426.1 HscLogicalSwitchModel

displayName	*string*	The switch name.
pSwitches	*table(string) = HscSimplePhysicalSwitchModel* see 2.426.1.1	A mapping between physical switch name and ports.
vni	*integer*	The VXLAN Network Identifier (VNI).

2.426.1.1 HscSimplePhysicalSwitchModel

ports	*list of string*	The physical ports which comprise the logical switch.

2.427 show hsc physical locator [...]

revision 3

locators	*list of IP address*	List of VTEP IP.

2.428 show hsc physical locator vni VNI [...]

revision 3

see 2.427

2.429 show hsc physical port [...]

revision 2

ports	*table(string) = HscPhysicalPortModel* see 2.429.1	A mapping between uuid and physical ports.

2.429.1 HscPhysicalPortModel

displayName	*string*	The port name.
switchName	*string*	The name of the physical switch.
uuid	*string*	The UUID of the port.
vlanBindings	*list of HscVlanBindingModel* see 2.429.1.1	The configured VLAN/VNI pairs.

2.429.1.1 HscVlanBindingModel

vlan	*integer*	The vLAN ID.
vni	*integer*	The VXLAN Network Identifier (VNI).

2.430 show hsc physical switch [...]

revision 2

ports	*table(string) = HscPhysicalPortsModel* see 2.429	A mapping between uuid and physical ports.
switches	*table(string) = HscPhysicalSwitchModel* see 2.430.1	A mapping between uuid and physical switch.

2.430.1 HscPhysicalSwitchModel

displayName	*string*	The switch name.
mgmtIps	*list of IP address*	The IP addresses of the management ports.

tunnelIps	*list of IP address*	The IP addresses of the tunnel end point.
uuid	*string*	The UUID of the switch.

2.431 show hsc status [...]

revision 3

[agent]	*HscAgentStatusModel* see 2.431.1	The status of the HSC agent.
[enableErrorReporting]	*boolean*	The error reporting state.
enabled	*boolean*	The configured state of the HSC service.
[fipsEnabled]	*boolean*	Whether FIPS mode is enabled for HTTPS algorithms.
[managers]	*list of OvsdbManagerModel* see 2.431.2	The currently configured managers and their connection state.
[ovsdbState]	*enum: Stopped | Disabled | Failed | Running | WaitingForConvergence | Started*	The state of the local OVSDB server.
[routingMode]	*enum: indirect | direct*	The routing mode.
[sslProfile]	*SslProfileModel* see 2.431.3	The configuration and status of the SSL profile.
tlsProtocol	*list of string*	Allowed TLS protocol versions.
[vtepFloodMode]	*enum: sendToAll | sendToAny*	The current Vtep flood mode.

2.431.1 HscAgentStatusModel

ovsdbConnection	*enum: Unknown | Connected | Disconnected*	The current agent to OVSDB server connection state.
state	*enum: Stopped | Running*	The current agent state.

2.431.2 OvsdbManagerModel

connected	*boolean*	Whether OVSDB server is currently connected to this manager.
manager	*string*	The manager connection information.
state	*string*	The current state of the connection to this manager.
uptime	*floating point number*	UTC Timestamp at time of manager connecting.

2.431.3 SslProfileModel

configured	*boolean*	Whether this profile is configured in SSL Manager.
name	*string*	SSL profile name.
[state]	*enum: valid | invalid | updating*	State of the SSL profile.

2.432 show hsc status ovsdb [...]

[lastErrorCode]	*string*	The last error encountered.
[lastFailedMsg]	*string*	The last failed message type.
[lastFailedMsgDetail]	*string*	The last failed message.
[lastFailedMsgTime]	*floating point number*	The time of the last failed message.
[lastMsgSent]	*string*	The type of the last OVSDB message sent.
[lastMsgSentDetail]	*string*	The last OVSDB message sent.
[lastStateChange]	*floating point number*	The last time the state changed.
[lastSuccessfulMsg]	*string*	The time of the last successful message.
[lastSuccessfulMsgDetail]	*string*	The last successful message sent.
[lastSuccessfulMsgTime]	*floating point number*	The time of the last successful message.
[numFailedMsg]	*integer*	The number of failed messages.
[numSuccessfulMsg]	*integer*	The number of successfull messages.
ovsdbState	*enum: Stopped | Started | Running*	The current state of the OVSDB server.
[responseTimeForLastMsg]	*floating point number*	The last message response time.
[sessionState]	*enum: Unknown | Connected | Disconnected*	The current agent state.
[timeOfLastMsg]	*floating point number*	The time the last message was sent.

2.433 show idprom (transceiver | interface) [...]

interfaces	*table(Interface name) = InterfacesTransceiverIdpromPages* see 2.433.1	Mapping between interface name and the eeprom.

2.433.1 InterfacesTransceiverIdpromPages

pages	*table(string) = InterfacesTransceiverIdpromData* see 2.433.1.1	Mapping of page name to register.

2.433.1.1 InterfacesTransceiverIdpromData

regWidth	*integer*	Eeprom register width in bits.
registers	*list of integer*	Eeprom data as an integer.

2.434 show igmp snooping querier [...]

vlans	*table(numeric string) = IgmpSnoopingVlanQuerier* see 2.434.1	Mapping vlan Id to the vlan querier info.

2.434.1 IgmpSnoopingVlanQuerier

igmpVersion	*enum: v2 | v1 | unknown | v3*	Version of IGMP snooping querier in the vlan.
querierAddress	*IPv4 address*	Address of IGMP querier in the Vlan.
querierInterface	*interface name*	Interface where the querier is located.
queryResponseInterval	*floating point number*	Effective maximum period a recipient can wait before responding with a membership report in seconds.

2.435 show igmp snooping querier counters [...]

vlans	*table(numeric string) = IgmpSnoopingVlanQuerier-Counters* see 2.435.1	Map Vlan id to Vlan counter info.

2.435.1 IgmpSnoopingVlanQuerierCounters

errorPacketReceived	*integer*	Number of error packets received.
igmpVersion	*enum: v2 | v1 | unknown | v3*	Version of IGMP snooping querier in the vlan.
operationalState	*enum: non-querier | querier*	Operational state of querier in the vlan.
otherPacketReceived	*integer*	Number of other packets received.
querierAddress	*IPv4 address*	Address of IGMP querier in the Vlan.
totalGeneralQueriesSent	*integer*	Total number of general queries sent.
v1QueriesReceived	*integer*	Number of v1 queries received.
v1QueriesSent	*integer*	Number of v1 general queries sent.
v1ReportReceived	*integer*	Number of v1 reports received.
v2LeaveReceived	*integer*	Number of v2 leaves received.
v2QueriesReceived	*integer*	Number of v2 queries received.
v2QueriesSent	*integer*	Number of v2 general queries sent.
v2ReportReceived	*integer*	Number of v2 reports received.
v3GSQueriesSent	*integer*	Number of v3 group specific queries sent.
v3GSSQueriesSent	*integer*	Number of v3 group and source specific queries sent.
v3QueriesReceived	*integer*	Number of v3 queries received.
v3QueriesSent	*integer*	Number of v3 general queries sent.
v3ReportReceived	*integer*	Number of v3 reports received.

2.436 show igmp snooping querier membership [...]

vlans	*table(numeric string) = IgmpSnoopingVlanQuerierMembership* see 2.436.1	Mapping vlan id with querier membership info.

2.436.1 IgmpSnoopingVlanQuerierMembership

addRandomDelayToTimer	*boolean*	True if random delay added to expiry timers.
[generalQueryExpiryTime]	*floating point number*	IGMP general query timer expiry in seconds.
[groupMembershipInterval]	*floating point number*	Period after which multicast group membership expires in seconds.
groups	*list of MulticastGroupInfo* see 2.436.1.2	Multicast group information.
[lastMemberQueryTime]	*floating point number*	Total time to send last member query messages in seconds.
[olderHostPresentInterval]	*floating point number*	Time-out for transitioning a group back to IGMPv3 mode once an older version report is received in seconds.
[otherQuerierExpiryTime]	*floating point number*	Other querier present timer expiry in seconds.
[otherQuerierPresentInterval]	*floating point number*	Period for which multicast router waits before becoming a querier in seconds.
querierStatus	*IgmpQuerierStatusVlan* see 2.436.1.1	Querier status information in the vlan.
robustness	*integer*	Number of queries sent to age out a port's membership in group.

2.436.1.1 IgmpQuerierStatusVlan

adminState	*enum: disabled | enabled*	State of IGMP querier in the Vlan.
[lastMemberQueryCount]	*integer*	Number of query messages that are sent before removing from group state and discontinue multicast transmissions in the vlan.
[lastMemberQueryInterval]	*floating point number*	Transmission frequency set for query messages to the group when last member on a particular group sends leave message in seconds.
operationalState	*enum: non-querier | querier*	Operational state of querier in the Vlan.
querierAddress	*IP address*	Ip Address of querier on the vlan.
querierExpiryTime	*floating point number*	Querier's expiry time in seconds.
queryInterval	*floating point number*	Period between IGMP Membership Query messages in seconds.
queryResponseInterval	*floating point number*	Effective maximum period a recipient can wait before responding with a Membership Report in seconds.
[startupQueryCount]	*integer*	Number of query messages that querier in the vlan sends during the startup query interval.

[startupQueryInterval]	*floating point number*	Period between query messages that querier in the vlan sends upon startup in seconds.
version	*enum: v2 | v1 | unknown | v3*	Version of IGMP querier on the Vlan.

2.436.1.2 MulticastGroupInfo

[creationTime]	*floating point number*	UTC time at which the group was created.
[expiryTime]	*floating point number*	UTC time at which the group will expire.
filterMode	*enum: exclude | include*	Filter mode of multicast group.
groupAddress	*IPv4 address*	Address of the multicast group.
igmpVersion	*enum: v2 | v1 | unknown | v3*	Version of IGMP snooping querier.
lastReporter	*IPv4 address*	The IP Address of the latest host who joined this group.
[sources]	*list of GroupSourceInfo* see 2.436.1.2.1	Sources belong to the group and its information.

2.436.1.2.1 GroupSourceInfo

[creationTime]	*floating point number*	UTC time at which this source was created.
expired	*boolean*	True if the source address expires.
[expiryTime]	*floating point number*	UTC time at which this source will expire.
sourceAddress	*IPv4 address*	Source address for multicast group.

2.437 show igmp snooping querier membership vlan VLAN_ID group IPADDR

sources	*list of GroupSourceInfo* see 2.436.1.2.1	Sources belong to the group and its information.

2.438 show igmp snooping querier status [...]

[adminState]	*enum: disabled | enabled*	State of global IGMP querier.
[lastMemberQueryCount]	*integer*	Number of query messages that are sent before removing from group state and discontinue multicast transmissions.
[lastMemberQueryInterval]	*floating point number*	Transmission frequency set for query messages to the group when last member on a particular group sends leave message in seconds.
[querierAddress]	*IPv4 address*	Address of global IGMP querier.
[querierExpiryTime]	*floating point number*	Querier's expiry time in seconds.
[queryInterval]	*floating point number*	Period between IGMP Membership Query messages sent from the global querier in seconds.
[queryResponseInterval]	*floating point number*	Effective maximum period a recipient can wait before responding with a Membership Report in seconds.

[robustness]	*integer*	Number of queries sent to age out a port's membership in group.
[startupQueryCount]	*integer*	Number of query messages that the querier sends during the startup query interval.
[startupQueryInterval]	*floating point number*	Period between query messages that the querier sends upon startup in seconds.
vlans	*table(numeric string) = IgmpQuerierStatusVlan* see 2.436.1.1	Map the vlan id with its querier info.

2.439 show installed-extensions

extensions	*table(string) = InstallationStatus* see 2.439.1	Maps the extension names to their statuses.

2.439.1 InstallationStatus

forced	*boolean*	Whether the installed extension is forced.

2.440 show interfaces (capabilities | hardware) | (capabilities | hardware (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfaceCapabilitiesData* see 2.440.1	A mapping of interfaces to their capabilities information.

2.440.1 InterfaceCapabilitiesData

[autoNegotiationClause28]	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 28 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[autoNegotiationClause37]	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 37 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[autoNegotiationClause73]	*AutonegClause73-Capabilities* see 2.440.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 73 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[coherentModes]	*list of CoherentMode* see 2.440.1.2	List of modulation and coherent error-correction capabilities for this interface.
[coherentTributaries] **	*list of CoherentTributary* see 2.440.1.3	List of tributaries which may be mapped at each line rate. **Since: 4.26.2
[hasAutoneg]	*boolean*	This interface supports speed auto negotiated.
modes	*list of EthIntfMode* see 2.440.1.4	List of link-mode capabilities for this interface.
rxFlowControl	*FlowControlCapabilities* see 2.440.1.5	The rx flow control capabilities of this interface.
[speedGroup]	*SpeedGroup* see 2.440.1.6	Speed Group Id and range of member interfaces.
switchModelName	*string*	Model name of the switch.
transceiverType	*string*	Type of the inserted transceiver.
txFlowControl	*FlowControlCapabilities* see 2.440.1.5	The tx flow control capabilities of this interface.

2.440.1.1 AutonegClause73Capabilities

consortium	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets capable of Consortium defined Clause 73 auto negotiation.
ieee	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets capable of IEEE defined Clause 73 auto negotiation.

2.440.1.1.1 EthSpeedLanesDuplex

[auto] **	*boolean*	This speed is auto negotiated. **Since: 4.23.1
[duplex]	*enum: half | full | unknown*	Duplex communication capabilities with this speed.
[laneCount]	*integer*	Number of lanes on the interface.
[speed]	*enum: Unknown | 200Gbps | Hg11Gbps | 1Gbps | 100Mbps | Hg21Gbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | Hg27Gbps | Hg106Gbps | Hg53Gbps | 10Gbps | 800Gbps | 100Gbps | 2p5Gbps | Hg42Gbps | 400Gbps | 5Gbps*	Interface speed.

2.440.1.2 CoherentMode

[defaultErrorCorrection]	*enum: RS G.709 | SD 25% | SD 15% | SC 7% | O-FEC | C-FEC | SD 20% | SD 25% BCH*	Default error correction.
[defaultLineRate] **	*enum: 100G | 200G | 400G | 300G*	Default line rate. **Since: 4.26.0
[defaultModulation]	*enum: DP-QPSK | 16QAM | none | 8QAM*	Default modulation type.
errorCorrections	*list of string*	List of coherent error-correction types available while using the corresponding modulation.
lineRates **	*list of string*	List of coherent line rate types available while using the corresponding modulation. **Since: 4.26.0
modulation	*enum: DP-QPSK | 16QAM | none | 8QAM*	Modulation type.

2.440.1.3 CoherentTributary

[defaultTributary]	*integer*	Default tributary.
[lineRate]	*enum: 100G | 200G | 400G | 300G*	Data rate of coherent optical link.
tributaries	*list of integer*	List of tributaries available while using the corresponding line rate.

2.440.1.4 EthIntfMode

[defaultErrorCorrection]	*enum: Disabled | FireCode | ReedSolomon | ReedSolomon544*	Default error correction.
[defaultLinkMode]	*boolean*	This is the default link mode for this interface.
[errorCorrections]	*list of string*	List of error-correction types capable of running with this speed.

`[link]`	*EthSpeedLanesDuplex* see 2.440.1.1.1	Interface speed and communication mode.
`[pllIncompatible]` **	*boolean*	This link mode is incompatible for this interface. **Since: 4.23.1

2.440.1.5 FlowControlCapabilities

`autoNegotiationPause`	*boolean*	True if we have pause capability for auto negotiation.
`flowControl`	*enum: Unknown* | *Incapable* | *Capable*	Flow control capability.

2.440.1.6 SpeedGroup

`groupId`	*string*	Speed group Id.
`intfList`	*list of Interface name*	List of speed group interface members.

2.441 show interfaces (capabilities | hardware default) | (capabilities | hardware default (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfaceCapabilities-DefaultData* see 2.441.1	A mapping of interfaces to their capabilities defaults information.

2.441.1 InterfaceCapabilitiesDefaultData

[autoNegotiationClause28]	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 28 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[autoNegotiationClause37]	*list of EthSpeedLanesDuplex* see 2.440.1.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 37 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[autoNegotiationClause73]	*AutonegClause73-Capabilities* see 2.440.1.1	List of speed/lane count/duplex triplets at which this interface can perform clause 73 auto negotiation. A missing entry indicates auto negotiation capabilities are unknown.
[coherentModes]	*list of CoherentMode* see 2.440.1.2	List of modulation and coherent error-correction capabilities for this interface.
[coherentTributaries] **	*list of CoherentTributary* see 2.440.1.3	List of tributaries which may be mapped at each line rate. **Since: 4.26.2
[hasAutoneg] **	*boolean*	This interface supports speed auto negotiated. **Since: 4.23.1
modes	*list of EthIntfMode* see 2.440.1.4	List of link-mode capabilities for this interface.
rxFlowControl	*FlowControlCapabilities* see 2.440.1.5	The rx flow control capabilities of this interface.
[speedGroup]	*SpeedGroup* see 2.440.1.6	Speed Group Id and range of member interfaces.
switchModelName	*string*	Model name of the switch.
transceiverType	*string*	Type of the inserted transceiver.
txFlowControl	*FlowControlCapabilities* see 2.440.1.5	The tx flow control capabilities of this interface.

2.442 show interfaces (counters bins) | (counters bins (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name)* = *InterfaceInOutBinsCounters* see 2.442.1	Mapping between an interface and counter bins.

2.442.1 InterfaceInOutBinsCounters

[inBinsCounters]	*InterfaceBinsCounters* see 2.442.1.1	counter bins for in frames.
[outBinsCounters]	*InterfaceBinsCounters* see 2.442.1.1	counter bins for out frames.

2.442.1.1 InterfaceBinsCounters

frames1024To1522Octet	*integer*	counter bin 1024-1522 bytes.
frames128To255Octet	*integer*	counter bin 128-255 bytes.
frames1523ToMaxOctet	*integer*	counter bin 1524-Max bytes.
frames256To511Octet	*integer*	counter bin 256-511 bytes.
frames512To1023Octet	*integer*	counter bin 512-1023 bytes.
frames64Octet	*integer*	counter bin 64 bytes.
frames65To127Octet	*integer*	counter bin 65-127 bytes.

2.443 show interfaces (counters errors) | (counters errors (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaceErrorCounters	*table(Interface name)* = *ErrorCounters* see 2.443.1	Mapping between an interface and error counters.

2.443.1 ErrorCounters

alignmentErrors	*integer*	Alignment errors.
fcsErrors	*integer*	FCS errors.
frameTooLongs	*integer*	Frame too long errors.
frameTooShorts	*integer*	Frame too short errors.
inErrors	*integer*	Input errors.
outErrors	*integer*	Output errors.
symbolErrors	*integer*	Symbol Errors.

2.444 show interfaces (counters half-duplex) | (counters half-duplex (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

`intfHalfDuplexErrorCounters`	*table(Interface name)* = *HalfDuplexErrorCounters* see 2.444.1	Mapping between an interface and half-duplex error counters.

2.444.1 HalfDuplexErrorCounters

`deferredTransmissions`	*integer*	Deferred transmissions.
`excessiveCollisions`	*integer*	Excessive collisions.
`lateCollisions`	*integer*	Late collisions.
`multipleCollisionFrames`	*integer*	Multiple collision frames.
`singleCollisionFrames`	*integer*	Single collision frames.

2.445 show interfaces (counters ingress acl drop) | (counters ingress acl drop (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

`interfaces`	*table(Interface name)* = *InterfacesCounterAclDrop* see 2.445.1	Mapping between an interface and ACL drop counters.

2.445.1 InterfacesCounterAclDrop

`aclDrops`	*integer*	Packets denied by an ACL.

2.446 show interfaces (error-correction) | (error-correction (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switch-card MODSLOT_SWITCHCARD)))) [...]

interfaceErrCorrStatuses	*table(Interface name) = InterfaceErrorCorrection-Status* see 2.446.1	Mapping between an interface and error correction information.

2.446.1 InterfaceErrorCorrectionStatus

[coherentEncoding-Configured] **	*enum: softDecision-Overhead15Percent | softDecisionOverhead25-PercentBch | default | concatenatedFec | softDecisionOverhead20-Percent | softDecisionOverhead25-Percent | openFec | reedSolomonG709 | staircaseOverhead7Percent*	Configured coherent error-correction encoding. **Since: 4.21.0
[concatenatedFecAvailable] **	*boolean*	True if C-FEC is available. **Since: 4.26.0
errCorrEncodingStatus	*enum: unknown | softDecisionOverhead15-Percent | softDecisionOverhead25-PercentBch | openFec | softDecisionOverhead25-Percent | reedSolomon | disabled | concatenatedFec | softDecisionOverhead20-Percent | fireCode | none | reedSolomonG709 | staircaseOverhead7Percent*	Current error correction encoding state.
fecEncodingConfigDisabled	*boolean*	True if error correction is disabled. When set, all other configured encoding values are ignored and should be False.
fecEncodingConfigFireCode-Enabled	*boolean*	True if Fire code (BASE-R) FEC, is enabled.
fecEncodingConfigReed-Solomon544Enabled **	*boolean*	True if Reed-Solomon FEC (544), is enabled. **Since: 4.21.0
fecEncodingConfigReed-SolomonEnabled	*boolean*	True if Reed-Solomon FEC (528), is enabled.
fecEncodingFireCode-Available	*boolean*	True if Fire code (BASE-R) FEC, is available.
fecEncodingReedSolomon544-Available **	*boolean*	True if RS-FEC-544 is available. **Since: 4.21.0
fecEncodingReedSolomon-Available	*boolean*	True if RS-FEC-528 is available.

`[openFecAvailable]` **	*boolean*	True if Open FEC is available. **Since: 4.25.1
`[reedSolomonG709Available]` **	*boolean*	G.709 Reed-Solomon FEC. **Since: 4.21.0
`[softDecisionOverhead15-PercentAvailable]` **	*boolean*	True if Soft Decision 15% overhead FEC is available. **Since: 4.21.0
`[softDecisionOverhead20-PercentAvailable]` **	*boolean*	True if Soft Decision 20% overhead FEC is available. **Since: 4.21.0
`[softDecisionOverhead25-PercentAvailable]` **	*boolean*	True if Soft Decision 25% overhead FEC is available. **Since: 4.21.0
`[softDecisionOverhead25-PercentBchAvailable]` **	*boolean*	True if Soft Decision 25% overhead BCH FEC is available. **Since: 4.21.0
`[staircaseOverhead7Percent-Available]` **	*boolean*	True if Staircase 7% overhead FEC is available. **Since: 4.21.0

2.447 show interfaces (flow-control | flowcontrol) | (flow-control | flowcontrol (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

see 2.368

2.448 show interfaces (l2 mtu) | (l2 mtu (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

`interfaceL2Mtus`	*table(Interface name) = L2Mtu* see 2.448.1	Mapping between an interface and errors.

2.448.1 L2Mtu

`l2MtuCfg`	*integer*	Configured L2 MTU (bytes).
`l2MtuStatus`	*integer*	L2 MTU Status (bytes).

2.449 show interfaces (phy detail) | (phy detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 2

interfacePhyStatuses	*table(Interface name)* = *InterfacePhyStatuses-Detailed* see 2.449.1	Collection of phy information, keyed by interface name.

2.449.1 InterfacePhyStatusesDetailed

[interfaceState]	*InterfaceState* see 2.449.1.1	State of the interface.
[macFaults]	*RsBaseR* see 2.449.1.2.40	MAC local/remote fault information.
[operSpeed]	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
phyStatuses	*list of PhyStatus* see 2.449.1.2	List of phy information for this interface.
[transceiver]	*Transceiver* see 2.449.1.49	Transceiver information.

2.449.1.1 InterfaceState

changes	*integer*	Number of interface state changes.
current	*enum: down | adminDown | up*	Current interface state.
lastChange	*floating point number*	Last time interface state changed.

2.449.1.2 PhyStatus

Could also be an instance of:
InternalGen2PhyIntfStatus 2.449.1.33
InternalGen3PhyIntfStatus 2.449.1.34
InternalPhyIntfStatus 2.449.1.38
RawPhyStatus 2.449.1.45
SwitchAsic400gPhyIntfStatus 2.449.1.37
B52ChipPhyStatus 2.449.1.5
B52LinePhyStatus 2.449.1.6
B52PhyStatus 2.449.1.7
B52SystemPhyStatus 2.449.1.8
BabbageChipPhyStatus 2.449.1.9
BabbageLinePhyStatus 2.449.1.10

BabbageSystemPhyStatus 2.449.1.11
Bcm54185LinePhyModel 2.449.1.12
Bcm54185PhyModel 2.449.1.13
Bcm541x2LinePhyModel 2.449.1.14
Bcm541x2PhyModel 2.449.1.15
Bcm87728ChipPhyStatus 2.449.1.16
Bcm87728LinePhyStatus 2.449.1.17
Bcm87728PhyStatus 2.449.1.18
Bcm87728SystemPhyStatus 2.449.1.19
BfnPhyModel 2.449.1.20
BfnPhyNiIntfStatus 2.449.1.21
CrosspointPhyStatus 2.449.1.25
DenaliLinePhyStatus 2.449.1.26
DenaliSystemPhyStatus 2.449.1.27
EnigmaLinePhyStatus 2.449.1.28
EnigmaSystemPhyStatus 2.449.1.29
EvoraLinePhyStatus 2.449.1.30
EvoraSystemPhyStatus 2.449.1.31
AlaskaLinePhyModel 2.449.1.3
AlaskaSystemPhyModel 2.449.1.4
CortinaHostPhyModel 2.449.1.23
CortinaLinePhyModel 2.449.1.24
PhyDs280ChipStatus 2.449.1.41
PhyDs280IntfStatus 2.449.1.42
PhyDs280LineIntfStatus 2.449.1.43
PhyDs280SystemIntfStatus 2.449.1.44
MillenioLinePhyStatus 2.449.1.39
MillenioSystemPhyStatus 2.449.1.40
BlackhawkPhyIntfStatus 2.449.1.22
FalconPhyIntfStatus 2.449.1.32
SfePhyNacIntfStatus 2.449.1.35
SfePhyNimIntfStatus 2.449.1.36
ShortfinLinePhyModel 2.449.1.46
ShortfinPhyModel 2.449.1.47
ShortfinSystemPhyModel 2.449.1.48

`[chip]`	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
`description`	*PhyDescription* see 2.449.1.2.24	Phy description.
`[fec]`	*Fec* see 2.449.1.2.1	FEC information.
`[pcs]`	*Pcs* see 2.449.1.2.8	PCS information.
`[pma]`	*Pma* see 2.449.1.2.25	PMA information.
`[preFecBer]`	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
`[rs]`	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
`[stale]`	*boolean*	Status of this phy might be stale.

2.449.1.2.1 Fec

Could also be an instance of:
FcFec 2.449.1.2.5
RsFec 2.449.1.2.6
RsFec544 2.449.1.2.7
BabbageRsFec 2.449.1.2.2
BfnPhyFcFecModel 2.449.1.2.3
BfnPhyRsFecModel 2.449.1.2.4

encoding	*enum: fireCode | reedSolomon*	The FEC encoding.

2.449.1.2.2 BabbageRsFec

[alignmentLock]	*RsFecAlignmentLock* see 2.449.1.2.2.22	FEC alignment lock status.
[codewordSize]	*enum: 544 | 528*	FEC codeword size.
[correctedCodewords]	*RsFecCorrectedCodewords* see 2.449.1.2.2.19	Number of corrected FEC codewords.
[correctedSymbolRate]	*floating point number*	FEC corrected symbol rate.
[correctedSymbols]	*table(numeric string) = RsFecCorrectedSymbols* see 2.449.1.2.2.20	Count of FEC corrected symbols per lane.
encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
[fifoStatus]	*FecFifoStatus* see 2.449.1.2.2.1	RS-FEC FIFO status.
[hiSer]	*RsFecHiSer* see 2.449.1.2.2.24	FEC high symbol error rate indication.
[laneAlignmentMarkerLock]	*table(numeric string) = RsFecAlignmentMarkerLock* see 2.449.1.2.2.23	FEC lane alignment marker lock.
[laneMap]	*table(numeric string) = integer*	FEC lane to PMA lane map.
[uncorrectedCodewords]	*RsFecUnCorrectedCodewords* see 2.449.1.2.2.21	Number of uncorrected FEC codewords.

2.449.1.2.2.1 FecFifoStatus

rxCnt	*integer*	FEC RX FIFO count value.
rxMax	*CountAndChangedStat* see 2.449.1.2.2.2	FEC RX FIFO max value.
rxMin	*CountAndChangedStat* see 2.449.1.2.2.2	FEC RX FIFO min value.
txCnt	*integer*	FEC TX FIFO count value.
txMax	*CountAndChangedStat* see 2.449.1.2.2.2	FEC TX FIFO max value.
txMin	*CountAndChangedStat* see 2.449.1.2.2.2	FEC TX FIFO min value.

2.449.1.2.2.2 CountAndChangedStat

Could also be an instance of:
PcsBipErrors 2.449.1.2.2.16
PcsLastErroredBlockCount 2.449.1.2.2.17
PcsLastHighBerCount 2.449.1.2.2.18
RsFecCorrectedCodewords 2.449.1.2.2.19
RsFecCorrectedSymbols 2.449.1.2.2.20
RsFecUnCorrectedCodewords 2.449.1.2.2.21
FecHistoBin 2.449.1.2.2.9
CredoPhyAdaptCnt 2.449.1.2.2.5
CredoPhyLinkLostCnt 2.449.1.2.2.6
CredoPhyLossOfSignalCnt 2.449.1.2.2.7
CredoPhyReAdaptCnt 2.449.1.2.2.8
FifoStatusEntry 2.449.1.2.2.10
BfnPhyLocalFaultCountModel 2.449.1.2.2.3
BfnPhyRemoteFaultCountModel 2.449.1.2.2.4
IdleErrorCount 2.449.1.2.2.11
PCSRXErrorCount 2.449.1.2.2.12
PCSRXGoodCount 2.449.1.2.2.13
PCSTXErrorCount 2.449.1.2.2.14
PCSTXGoodCount 2.449.1.2.2.15

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.3 BfnPhyLocalFaultCountModel

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.4 BfnPhyRemoteFaultCountModel

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.5 CredoPhyAdaptCnt

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.6 CredoPhyLinkLostCnt

changes	*integer*	The number of times the count was non-zero.

lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.7 CredoPhyLossOfSignalCnt

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.8 CredoPhyReAdaptCnt

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.9 FecHistoBin

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.10 FifoStatusEntry

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.11 IdleErrorCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.12 PCSRXErrorCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.13 PCSRXGoodCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.14 PCSTXErrorCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.15 PCSTXGoodCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.16 PcsBipErrors

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.17 PcsLastErroredBlockCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.18 PcsLastHighBerCount

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.19 RsFecCorrectedCodewords

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.

value	*integer*	Last non-zero counter value.

2.449.1.2.2.20 RsFecCorrectedSymbols

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.21 RsFecUnCorrectedCodewords

changes	*integer*	The number of times the count was non-zero.
lastChange	*floating point number*	The last time the count was non-zero.
value	*integer*	Last non-zero counter value.

2.449.1.2.2.22 RsFecAlignmentLock

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.2.23 RsFecAlignmentMarkerLock

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.2.24 RsFecHiSer

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.3 BfnPhyFcFecModel

encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
perLaneCorrectedFecBlocks	*table(numeric string) = CountAndChangedStat* see 2.449.1.2.2.2	Count of corrected FEC blocks per FEC lane.

perLaneUncorrectedFecBlocks	*table(numeric string) = CountAndChangedStat* see 2.449.1.2.2.2	Count of uncorrected FEC blocks per FEC lane.

2.449.1.2.4 BfnPhyRsFecModel

[alignmentLock]	*RsFecAlignmentLock* see 2.449.1.2.2.22	FEC alignment lock status.
[codewordSize]	*enum: 544 | 528*	FEC codeword size.
[correctedCodewords]	*RsFecCorrectedCodewords* see 2.449.1.2.2.19	Number of corrected FEC codewords.
[correctedSymbolRate] **	*floating point number*	FEC corrected symbol rate. **Since: 4.24.1
[correctedSymbols]	*table(numeric string) = RsFecCorrectedSymbols* see 2.449.1.2.2.20	Count of FEC corrected symbols per lane.
encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
[hiSer]	*RsFecHiSer* see 2.449.1.2.2.24	FEC high symbol error rate indication.
[laneAlignmentMarkerLock]	*table(numeric string) = RsFecAlignmentMarkerLock* see 2.449.1.2.2.23	FEC lane alignment marker lock.
[laneMap]	*table(numeric string) = integer*	FEC lane to PMA lane map.
[uncorrectedCodewords]	*RsFecUnCorrectedCodewords* see 2.449.1.2.2.21	Number of uncorrected FEC codewords.

2.449.1.2.5 FcFec

encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
perLaneCorrectedFecBlocks	*table(numeric string) = CountAndChangedStat* see 2.449.1.2.2.2	Count of corrected FEC blocks per FEC lane.
perLaneUncorrectedFecBlocks	*table(numeric string) = CountAndChangedStat* see 2.449.1.2.2.2	Count of uncorrected FEC blocks per FEC lane.

2.449.1.2.6 RsFec

[alignmentLock]	*RsFecAlignmentLock* see 2.449.1.2.2.22	FEC alignment lock status.
[codewordSize]	*enum: 544 | 528*	FEC codeword size.
[correctedCodewords]	*RsFecCorrectedCodewords* see 2.449.1.2.2.19	Number of corrected FEC codewords.
[correctedSymbolRate] **	*floating point number*	FEC corrected symbol rate. **Since: 4.24.1
[correctedSymbols]	*table(numeric string) = RsFecCorrectedSymbols* see 2.449.1.2.2.20	Count of FEC corrected symbols per lane.

encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
[hiSer]	*RsFecHiSer* see 2.449.1.2.2.24	FEC high symbol error rate indication.
[laneAlignmentMarkerLock]	*table(numeric string) = RsFecAlignmentMarkerLock* see 2.449.1.2.2.23	FEC lane alignment marker lock.
[laneMap]	*table(numeric string) = integer*	FEC lane to PMA lane map.
[uncorrectedCodewords]	*RsFecUnCorrectedCodewords* see 2.449.1.2.2.21	Number of uncorrected FEC codewords.

2.449.1.2.7 RsFec544

[alignmentLock]	*RsFecAlignmentLock* see 2.449.1.2.2.22	FEC alignment lock status.
[codewordSize]	*enum: 544 | 528*	FEC codeword size.
[correctedCodewords]	*RsFecCorrectedCodewords* see 2.449.1.2.2.19	Number of corrected FEC codewords.
[correctedSymbolRate] **	*floating point number*	FEC corrected symbol rate. **Since: 4.24.1
[correctedSymbols]	*table(numeric string) = RsFecCorrectedSymbols* see 2.449.1.2.2.20	Count of FEC corrected symbols per lane.
encoding	*enum: fireCode | reedSolomon*	The FEC encoding.
[hiSer]	*RsFecHiSer* see 2.449.1.2.2.24	FEC high symbol error rate indication.
[laneAlignmentMarkerLock]	*table(numeric string) = RsFecAlignmentMarkerLock* see 2.449.1.2.2.23	FEC lane alignment marker lock.
[laneMap]	*table(numeric string) = integer*	FEC lane to PMA lane map.
[uncorrectedCodewords]	*RsFecUnCorrectedCodewords* see 2.449.1.2.2.21	Number of uncorrected FEC codewords.

2.449.1.2.8 Pcs

Could also be an instance of:
MultiLanePcsBaseR 2.449.1.2.17
PcsBaseR 2.449.1.2.18
PcsClause45 2.449.1.2.19
B52MultiLanePcs 2.449.1.2.9
Bcm54185PcsModel 2.449.1.2.10
Bcm541x2PcsModel 2.449.1.2.11
BfnPhyPcsModel 2.449.1.2.12
DenaliPcs 2.449.1.2.14
EnigmaMultiLanePcs 2.449.1.2.15
EvoraMultiLanePcs 2.449.1.2.16
CortinaPcsModel 2.449.1.2.13
ShortfinPcsModel 2.449.1.2.20

2.449.1.2.9 B52MultiLanePcs

[bipErrors]	*table(numeric string) = PcsBipErrors* see 2.449.1.2.2.16	PCS BIP errors.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[laneAlignmentMarkerLock]	*table(numeric string) = PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS lane alignment marker lock.
[laneBlockLock]	*table(numeric string) = PcsBlockLock* see 2.449.1.2.9.2	PCS lane block lock.
[laneDeskew]	*table(numeric string) = PcsDeskew* see 2.449.1.2.9.3	PCS lane deskew status.
[laneMap]	*table(numeric string) = integer*	Local PCS lane to peer PCS lane map.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.9.1 PcsAlignmentMarkerLock

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.2 PcsBlockLock

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.3 PcsDeskew

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.4 PcsHighBer

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.5 PcsLinkStatus

changes	*integer*	The number of times the link status changed.
lastChange	*floating point number*	The last time the link status changed.
value	*enum: down | up*	The status of the link.

2.449.1.2.9.6 PcsRxFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.7 PcsRxLpi

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.9.8 PcsTxFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.10 Bcm54185PcsModel

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.

[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.11 Bcm541x2PcsModel

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.12 BfnPhyPcsModel

[bipErrors]	*table(numeric string) = PcsBipErrors* see 2.449.1.2.2.16	PCS BIP errors.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[laneAlignmentMarkerLock]	*table(numeric string) = PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS lane alignment marker lock.
[laneBlockLock]	*table(numeric string) = PcsBlockLock* see 2.449.1.2.9.2	PCS lane block lock.

[laneDeskew]	*table(numeric string) = PcsDeskew* see 2.449.1.2.9.3	PCS lane deskew status.
[laneMap]	*table(numeric string) = integer*	Local PCS lane to peer PCS lane map.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.13 CortinaPcsModel

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.14 DenaliPcs

alignmentMarkerLock	*PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS alignment marker lock.
bipErrorsDetected	*PcsBipErrorsDetected* see 2.449.1.2.14.1	PCS BIP error detected.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.

[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
lossOfAlignment	*PcsLossOfAlignment* see 2.449.1.2.14.3	PCS loss of alignment.
pcsErroredBlocksDetected	*PcsErroredBlocksDetected* see 2.449.1.2.14.2	PCS errored blocks detected.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.14.1 PcsBipErrorsDetected

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.14.2 PcsErroredBlocksDetected

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.14.3 PcsLossOfAlignment

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.15 EnigmaMultiLanePcs

[bipErrors]	*table(numeric string) = PcsBipErrors* see 2.449.1.2.2.16	PCS BIP errors.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.

[laneAlignmentMarkerLock]	*table(numeric string) = PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS lane alignment marker lock.
[laneBlockLock]	*table(numeric string) = PcsBlockLock* see 2.449.1.2.9.2	PCS lane block lock.
[laneDeskew]	*table(numeric string) = PcsDeskew* see 2.449.1.2.9.3	PCS lane deskew status.
[laneMap]	*table(numeric string) = integer*	Local PCS lane to peer PCS lane map.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.16 EvoraMultiLanePcs

[bipErrors]	*table(numeric string) = PcsBipErrors* see 2.449.1.2.2.16	PCS BIP errors.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
deskewAchieved	*PcsDeskewAchieved* see 2.449.1.2.16.1	Whether or not deskew has been achieved.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[laneAlignmentMarkerLock]	*table(numeric string) = PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS lane alignment marker lock.
[laneBlockLock]	*table(numeric string) = PcsBlockLock* see 2.449.1.2.9.2	PCS lane block lock.
[laneDeskew]	*table(numeric string) = PcsDeskew* see 2.449.1.2.9.3	PCS lane deskew status.
[laneMap]	*table(numeric string) = integer*	Local PCS lane to peer PCS lane map.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.

linkInterrupted	*PcsLinkInterrupted* see 2.449.1.2.16.2	Whether or not link has been interrupted.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.16.1 PcsDeskewAchieved

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.16.2 PcsLinkInterrupted

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.17 MultiLanePcsBaseR

[bipErrors]	*table(numeric string) = PcsBipErrors* see 2.449.1.2.2.16	PCS BIP errors.
[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[laneAlignmentMarkerLock]	*table(numeric string) = PcsAlignmentMarkerLock* see 2.449.1.2.9.1	PCS lane alignment marker lock.
[laneBlockLock]	*table(numeric string) = PcsBlockLock* see 2.449.1.2.9.2	PCS lane block lock.
[laneDeskew]	*table(numeric string) = PcsDeskew* see 2.449.1.2.9.3	PCS lane deskew status.
[laneMap]	*table(numeric string) = integer*	Local PCS lane to peer PCS lane map.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.

[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.18 PcsBaseR

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[deskew]	*PcsDeskew* see 2.449.1.2.9.3	PCS deskew status.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.19 PcsClause45

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.20 ShortfinPcsModel

[blockLock]	*PcsBlockLock* see 2.449.1.2.9.2	PCS block lock.
[highBer]	*PcsHighBer* see 2.449.1.2.9.4	PCS high BER status.
[lastErroredBlockCount]	*PcsLastErroredBlockCount* see 2.449.1.2.2.17	PCS last errored block count.
[lastHighBerCount]	*PcsLastHighBerCount* see 2.449.1.2.2.18	PCS last high BER count.
[linkStatus]	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[rxFault]	*PcsRxFault* see 2.449.1.2.9.6	PCS receive fault.
[rxLpi]	*PcsRxLpi* see 2.449.1.2.9.7	PCS has received LPI signaling.
[txFault]	*PcsTxFault* see 2.449.1.2.9.8	PCS transmit fault.

2.449.1.2.21 PhyChipStatus

Could also be an instance of:
ChipStatus 2.449.1.2.22
ChipStatus 2.449.1.2.23

[hwRev]	*string*	PHY hardware revision.
model	*integer*	Model number.
[modelName]	*string*	PHY model name.
oui	*integer*	Organizationally Unique Identifier (OUI).
rev	*integer*	Revision number.

2.449.1.2.22 ChipStatus

[firmwareRev]	*string*	PHY firmware revision.
[hwRev]	*string*	PHY hardware revision.
model	*integer*	Model number.
[modelName]	*string*	PHY model name.
oui	*integer*	Organizationally Unique Identifier (OUI).
rev	*integer*	Revision number.

2.449.1.2.23 ChipStatus

[availableFirmwareRev] **	*string*	PHY firmware revision available in current EOS image. **Since: 4.23.1
[firmwareRev] **	*string*	PHY firmware revision. **Since: 4.23.1
[firmwareStatus] **	*PhyFirmwareStatus* see 2.449.1.2.23.1	PHY firmware status information. **Since: 4.27.2
[hwRev]	*string*	PHY hardware revision.
model	*integer*	Model number.

[modelName]	*string*	PHY model name.
oui	*integer*	Organizationally Unique Identifier (OUI).
rev	*integer*	Revision number.

2.449.1.2.23.1 PhyFirmwareStatus

changes	*integer*	Number of PHY firmware status changes.
lastChange	*floating point number*	Last time PHY firmware status changed.
value	*enum: error | ok | unknown*	Current PHY firmware status.

2.449.1.2.24 PhyDescription

[location]	*enum: system | line*	Location of phy within phy chip.
phyChipName	*string*	Name of the phy chip.
[position] **	*integer*	Position of the phy chip in topology. **Since: 4.29.1

2.449.1.2.25 Pma

Could also be an instance of:
PmaBaseR 2.449.1.2.33
PmaBaseT 2.449.1.2.35
PmaClause45 2.449.1.2.36
PmaGen2 2.449.1.2.34
Bcm54185PmaModel 2.449.1.2.26
Bcm541x2PmaModel 2.449.1.2.27
BfnPhyPmaModel 2.449.1.2.28
DenaliPma 2.449.1.2.30
EnigmaPma 2.449.1.2.31
EvoraPma 2.449.1.2.32
CortinaPmaModel 2.449.1.2.29
ShortfinPmaModel 2.449.1.2.37

2.449.1.2.26 Bcm54185PmaModel

[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.26.1 PmaLinkStatus

changes	*integer*	The number of times the link status changed.
lastChange	*floating point number*	The last time the link status changed.
value	*enum: down | up*	The status of the link.

2.449.1.2.26.2 PmaRxFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.26.3 PmaSignalDetect

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.26.4 PmaTxFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.27 Bcm541x2PmaModel

[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.28 BfnPhyPmaModel

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.

[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.29 CortinaPmaModel

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.30 DenaliPma

adcOutput	*AdcOutput* see 2.449.1.2.30.1	Whether or not ADC output is set.
adcReady	*AdcReady* see 2.449.1.2.30.2	Whether or not ADC is ready.
carrierFrequencyOffset	*CarrierFrequencyOffset* see 2.449.1.2.30.3	Receiver carrier frequency offset in MHz.
chromaticDispersion	*ChromaticDispersion* see 2.449.1.2.30.4	Chromatic dispersion in ps/nm.
dacReady	*DacReady* see 2.449.1.2.30.5	Whether or not DAC is ready.
differentialGroupDelay	*DifferentialGroupDelay* see 2.449.1.2.30.6	Differential group delay in ps.
dispersionLockSet	*DispersionLockSet* see 2.449.1.2.30.7	Whether or not the DSP found a good dispersion compensation setting.
hiAgcError	*HiAgcError* see 2.449.1.2.30.9	Sampled HI quad error filter AGC error.
hqAgcError	*HqAgcError* see 2.449.1.2.30.10	Sampled HQ quad error filter AGC error.
linkQFactor	*LinkQFactor* see 2.449.1.2.30.11	Link Q-factor in dB.
preFecBitErrorRate	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
rollOff	*RollOff* see 2.449.1.2.30.13	Amount of excess bandwidth of the filter, as a fraction of frequency.

rxAsicSubsystemConfigured	*RxAsicSubsystems-Configured* see 2.449.1.2.30.14	Whether or not the RX ASIC subsystems has been configured.
rxClockConfigured	*RxClockConfigured* see 2.449.1.2.30.15	Whether or not the RX clock has been configured.
rxDelayH	*RxDelayH* see 2.449.1.2.30.16	HQ delay in ps.
rxDelayV	*RxDelayV* see 2.449.1.2.30.17	VQ delay in ps.
rxDemodLockSet	*RxDemodLockSet* see 2.449.1.2.30.18	Whether or not there is RX demodulator lock.
signalNoiseRatio	*SignalNoiseRatio* see 2.449.1.2.30.19	Signal-to-noise ratio in dB.
txAsicSubsystemConfigured	*TxAsicSubsystems-Configured* see 2.449.1.2.30.20	Whether or not the TX ASIC subsystems has been configured.
txClockConfigured	*TxClockConfigured* see 2.449.1.2.30.21	Whether or not the TX clock has been configured.
txDelayH	*TxDelayH* see 2.449.1.2.30.22	Delay for launching the signal on HQ.
txDelayV	*TxDelayV* see 2.449.1.2.30.23	Delay for launching the signal on VQ.
uncorrectedCodewords	*FecUnCorrectedCodewords* see 2.449.1.2.30.8	Number of uncorrected FEC code words.
viAgcError	*ViAgcError* see 2.449.1.2.30.24	Sampled VI quad error filter AGC error.
vqAgcError	*VqAgcError* see 2.449.1.2.30.25	Sampled VQ quad error filter AGC error.

2.449.1.2.30.1 AdcOutput

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.2 AdcReady

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.3 CarrierFrequencyOffset

value	*integer*	The value of the attribute.

2.449.1.2.30.4 ChromaticDispersion

value	*integer*	The value of the attribute.

2.449.1.2.30.5 DacReady

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.6 DifferentialGroupDelay

value	*integer*	The value of the attribute.

2.449.1.2.30.7 DispersionLockSet

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.8 FecUnCorrectedCodewords

changes	*integer*	The number of times the value changed.
lastChange	*floating point number*	The last time the value changed.
value	*integer*	The value of the attribute.

2.449.1.2.30.9 HiAgcError

value	*integer*	The value of the attribute.

2.449.1.2.30.10 HqAgcError

value	*integer*	The value of the attribute.

2.449.1.2.30.11 LinkQFactor

value	*floating point number*	The value of the attribute.

2.449.1.2.30.12 PreFecBitError

value	*floating point number*	Pre-FEC bit error rate.

2.449.1.2.30.13 RollOff

value	*floating point number*	The value of the attribute.

2.449.1.2.30.14 RxAsicSubsystemsConfigured

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.15 RxClockConfigured

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.16 RxDelayH

value	*floating point number*	The value of the attribute.

2.449.1.2.30.17 RxDelayV

value	*floating point number*	The value of the attribute.

2.449.1.2.30.18 RxDemodLockSet

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.19 SignalNoiseRatio

value	*floating point number*	The value of the attribute.

2.449.1.2.30.20 TxAsicSubsystemsConfigured

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.21 TxClockConfigured

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.30.22 TxDelayH

value	*floating point number*	The value of the attribute.

2.449.1.2.30.23 TxDelayV

value	*floating point number*	The value of the attribute.

2.449.1.2.30.24 ViAgcError

value	*integer*	The value of the attribute.

2.449.1.2.30.25 VqAgcError

value	*integer*	The value of the attribute.

2.449.1.2.31 EnigmaPma

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.32 EvoraPma

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.

[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.33 PmaBaseR

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.34 PmaGen2

[laneLinkStatus]	*table(numeric string) = PmaLinkStatus* see 2.449.1.2.26.1	PMA lane link status.
[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[laneTxStatus]	*table(numeric string) = PmaTxStatus* see 2.449.1.2.34.1	PMA per lane transmit status.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.
[txStatus]	*PmaTxStatus* see 2.449.1.2.34.1	PMA lane aggregate transmit status.

2.449.1.2.34.1 PmaTxStatus

changes	*integer*	The number of times the transmit status changed.

lastChange	*floating point number*	The last time the transmit status changed.
value	*enum: off | on*	The transmit status.

2.449.1.2.35 PmaBaseT

[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkPartnerInfoValid]	*LpInformationValid* see 2.449.1.2.35.1	Whether or not link partner information is valid.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.35.1 LpInformationValid

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.36 PmaClause45

[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.
[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.37 ShortfinPmaModel

[laneSignalDetect]	*table(numeric string) = PmaSignalDetect* see 2.449.1.2.26.3	PMA lane signal detect.
[linkStatus]	*PmaLinkStatus* see 2.449.1.2.26.1	PMA link status.
[rxFault]	*PmaRxFault* see 2.449.1.2.26.2	PMA receive fault.

[signalDetect]	*PmaSignalDetect* see 2.449.1.2.26.3	PMA signal detect.
[txFault]	*PmaTxFault* see 2.449.1.2.26.4	PMA transmit fault.

2.449.1.2.38 Rs

Could also be an instance of:
RsBaseR 2.449.1.2.40
BfnPhyRsModel 2.449.1.2.39

2.449.1.2.39 BfnPhyRsModel

[lfCnt]	*BfnPhyLocalFaultCount-Model* see 2.449.1.2.2.3	BfnPhy Local Fault Count.
[localFault]	*RsLocalFault* see 2.449.1.2.39.1	MAC Local fault.
[remoteFault]	*RsRemoteFault* see 2.449.1.2.39.2	MAC remote fault.
[rfCnt]	*BfnPhyRemoteFaultCount-Model* see 2.449.1.2.2.4	BfnPhy Remote Fault Count.

2.449.1.2.39.1 RsLocalFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.39.2 RsRemoteFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.2.40 RsBaseR

[localFault]	*RsLocalFault* see 2.449.1.2.39.1	MAC Local fault.
[remoteFault]	*RsRemoteFault* see 2.449.1.2.39.2	MAC remote fault.

2.449.1.3 AlaskaLinePhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
downShift	*DownShift* see 2.449.1.3.1	Downshift.
dtePower	*DtePower* see 2.449.1.3.2	DTE Power.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
fifoIdleDeleted	*FifoIdleDeleted* see 2.449.1.3.3	FifoIdleDeleted.
fifoIdleInserted	*FifoIdleInserted* see 2.449.1.3.4	FifoIdleInserted.
fifoUnderflow	*FifoUnderflow* see 2.449.1.3.5	FifoUnderflow.
[idleErrorCount]	*IdleErrorCount* see 2.449.1.2.2.11	IdleErrorCount.
jabber	*Jabber* see 2.449.1.3.6	Jabber.
linkStatus	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[localReceiverOk]	*LocalReceiverOk* see 2.449.1.3.7	LocalReceiverOk.
[masterSlaveFault]	*MasterSlaveFault* see 2.449.1.3.8	MasterSlaveFault.
mdi	*Mdi* see 2.449.1.3.9	MDI.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
polarity	*Polarity* see 2.449.1.3.10	Polarity.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[remoteReceiverOk]	*RemoteReceiverOk* see 2.449.1.3.11	RemoteReceiverOk.
resolved	*Resolved* see 2.449.1.3.12	Speed/duplex resolution.
[resolvedToMaster]	*ResolvedToMaster* see 2.449.1.3.13	ResolvedToMaster.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.3.1 DownShift

changes	*integer*	The number of times the condition changed.

lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.2 DtePower

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.3 FifoIdleDeleted

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.4 FifoIdleInserted

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.5 FifoUnderflow

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.6 Jabber

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.7 LocalReceiverOk

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.

value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.8 MasterSlaveFault

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.9 Mdi

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.10 Polarity

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.11 RemoteReceiverOk

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.12 Resolved

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.3.13 ResolvedToMaster

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.4 AlaskaSystemPhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
linkStatus	*PcsLinkStatus* see 2.449.1.2.9.5	PCS link status.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
qsgmiiAnegBypass	*QsgmiiAnegBypass* see 2.449.1.4.1	QsgmiiAnegBypass.
qsgmiiSyncStatus	*QsgmiiSyncStatus* see 2.449.1.4.2	QsgmiiSyncStatus.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.4.1 QsgmiiAnegBypass

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.4.2 QsgmiiSyncStatus

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.5 B52ChipPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
firmwareRev	*string*	PHY firmware revision.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.

[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.6 B52LinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
lanes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes lane to SerDes diagnostic information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[topPllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Maximum voltage control oscillator capacitor of phase locked loop over all lanes.

2.449.1.6.1 CredoPhyPllVcoCap

[rxPllCap]	*integer*	Rx VcoCap of Pll, encoded as 0 to 127.
[txPllCap]	*integer*	Tx VcoCap of Pll, encoded as 0 to 127.

2.449.1.6.2 CredoPhySerdes

[adaptCnt]	*CredoPhyAdaptCnt* see 2.449.1.2.2.5	Firmware adaptation counter.
[adaptCompleted]	*CredoPhyAdaptCompleted* see 2.449.1.6.2.1	Firmware adaptation completed.
[channelEst]	*floating point number*	Channel estimation.
[ctleGain]	*list of integer*	SerDes lane Continuous Time Linear Equalizer gain values.
[ctlePeaking]	*list of integer*	SerDes lane Continuous Time Linear Equalizer peaking values.
[dfeTaps]	*list of floating point number*	SerDes lane Decision Feedback Equalizer taps, ordered as main tap first tap, second tap, ... , last tap and initial tap.

[eyeMargins]	*list of floating point number*	Eye margin in mV, in case of multi-bit encodings, ordered by top to bottom.
[highFreqTrans]	*integer*	High frequency transition count after equalization.
[linkLostCnt]	*CredoPhyLinkLostCnt* see 2.449.1.2.2.6	Firmware link lost counter.
linkTrainingEnable **	*boolean*	Tx link training enable. **Since: 4.24.2
linkTrainingStatus **	*enum: link-ready | invalid | in-training | init | complete | train-remote | failed | train-local*	Link training Status. **Since: 4.24.2
[losCnt] **	*CredoPhyLossOfSignalCnt* see 2.449.1.2.2.7	Firmware lost of signal counter. **Since: 4.26.2
[pllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Voltage control oscillator capacitor of phase locked loop.
[reAdaptCnt]	*CredoPhyReAdaptCnt* see 2.449.1.2.2.8	Firmware re-adaptation counter.
[rxDacGain]	*integer*	RX digital sampler gain value.
rxFreqPPM	*integer*	Frequency offset of local reference clock with respect to the frequency of received data in parts per million of Hz.
[signalAmplitude]	*integer*	Signal envelope to represent the amplitude of signal after equalization, encoded as 0 to 63.
[txTaps]	*CredoPhyTxTaps* see 2.449.1.6.2.2	SerDes lane TX pre-emphasis taps.

2.449.1.6.2.1 CredoPhyAdaptCompleted

changes	*integer*	The number of times the condition changed.
lastChange	*floating point number*	The last time the condition changed.
value	*boolean*	Whether or not a condition is true or present.

2.449.1.6.2.2 CredoPhyTxTaps

main	*integer*	The main cursor tap.
post1	*integer*	The first post cursor tap.
post2	*integer*	The second post cursor tap.
pre1	*integer*	The first pre cursor tap.
pre2	*integer*	The second pre cursor tap.

2.449.1.7 B52PhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.

[fec]	*Fec* see 2.449.1.2.1	FEC information.
lanes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes lane to SerDes diagnostic information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[topPllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Maximum voltage control oscillator capacitor of phase locked loop over all lanes.

2.449.1.8 B52SystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
lanes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes lane to SerDes diagnostic information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[topPllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Maximum voltage control oscillator capacitor of phase locked loop over all lanes.

2.449.1.9 BabbageChipPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
firmwareRev	*string*	PHY firmware revision.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.

[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.10 BabbageLinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
lanes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes network lane to CRT50216/CRT55321 SerDes info.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[topPllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Maximum voltage control oscillator capacitor of phase locked loop over all lanes.

2.449.1.11 BabbageSystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
lanes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes system lane to CRT50216/CRT55321 SerDes info.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[topPllVcoCap]	*CredoPhyPllVcoCap* see 2.449.1.6.1	Maximum voltage control oscillator capacitor of phase locked loop over all lanes.

2.449.1.12 Bcm54185LinePhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.13 Bcm54185PhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
chipStatus	*PhyChipStatus* see 2.449.1.2.21	Bcm54185 chip status.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
speed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.14 Bcm541x2LinePhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.

[fec]	*Fec* see 2.449.1.2.1	FEC information.
mdiStatus **	*MdiCrossoverModel* see 2.449.1.14.1	MDI/MDI-X status. **Since: 4.30.0
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.14.1 MdiCrossoverModel

changes	*integer*	Number of MDI state changes.
lastChange	*floating point number*	Last time MDI state changed.
mdiState	*enum: mdix | mdi*	MDI/MDI-X layout.

2.449.1.15 Bcm541x2PhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
chipStatus	*PhyChipStatus* see 2.449.1.2.21	Bcm541x2 chip status.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
speed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.16 Bcm87728ChipPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[firmwareRev] **	*string*	PHY firmware revision. **Since: 4.30.1
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.17 Bcm87728LinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes **	*BlackhawkLtOptionalPhySerdesModel* see 2.449.1.17.1	The SerDes statistics. **Since: 4.30.1
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.17.1 BlackhawkLtOptionalPhySerdesModel

serdesStats	*table(numeric string) = BlackhawkPhySerdesLaneModelBase* see 2.449.1.17.1.1	The serdes statistics.
sortedSerdesStats	*list of integer*	Sorted Serdes statistics.

2.449.1.17.1.1 BlackhawkPhySerdesLaneModelBase

Could also be an instance of:
BlackhawkLtOptionalPhySerdesLaneModel 2.449.1.17.1.2
BlackhawkPhySerdesLaneModel 2.449.1.17.1.3

brPdEnable	*enum: BR | OS*	Baud rate phase detector enable.
[brPdForced] **	*boolean*	Whether the BR CDR is forced. **Since: **currentRelease**
[dfeTaps]	*table(numeric string) = integer*	Serdes lane Decision Feedback Equalizer (DFE) taps.
[eyeLeft]	*integer*	Horizontal left eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
[eyeLower]	*integer*	Vertical lower eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
[eyeRight]	*integer*	Horizontal right eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
[eyeUpper]	*integer*	Vertical upper eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
highFrequencyPeakingFilter	*integer*	High frequency peaking filter control.
lowFrequncyPeakingFilter	*integer*	Low frequency peaking filter control.
m1SlicerThreshold	*integer*	Vertical threshold voltage of m1 slicer (mV), 'channel loss hint' in PAM4 mode should always be 0.
mainPeakingFilter	*integer*	Peaking filter main settings.
[osPdForced] **	*boolean*	Whether the OS CDR is forced. **Since: **currentRelease**
p1SlicerThreshold	*integer*	Vertical threshold voltage of p1 slicer (mV).
rxPamMode	*enum: PAM4_NS | PAM4_ES | NRZ*	Receiver PAM4 mode.
[rxPpm]	*integer*	Frequency offset of local reference clock with respect to the frequency of received data in ppm.
[txPpm]	*integer*	Frequency offset of local reference clock with respect to the transmit data in ppm.
[txTaps]	*table(numeric string) = integer*	Serdes lane TX pre-emphasis taps.
[vga]	*integer*	Variable gain amplifier value.

2.449.1.17.1.2 BlackhawkLtOptionalPhySerdesLaneModel

brPdEnable	*enum: BR | OS*	Baud rate phase detector enable.
[brPdForced] **	*boolean*	Whether the BR CDR is forced. **Since: **currentRelease**
[dfeTaps]	*table(numeric string) = integer*	Serdes lane Decision Feedback Equalizer (DFE) taps.
[eyeLeft]	*integer*	Horizontal left eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
[eyeLower]	*integer*	Vertical lower eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
[eyeRight]	*integer*	Horizontal right eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.

[eyeUpper]	*integer*	Vertical upper eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
highFrequencyPeakingFilter	*integer*	High frequency peaking filter control.
[linkTrainingEnable]	*boolean*	Tx link training enable.
[linkTrainingStatus]	*enum: link-ready* | *invalid* | *in-training* | *init* | *complete* | *train-remote* | *failed* | *train-local*	Link training status.
lowFrequncyPeakingFilter	*integer*	Low frequency peaking filter control.
m1SlicerThreshold	*integer*	Vertical threshold voltage of m1 slicer (mV), 'channel loss hint' in PAM4 mode should always be 0.
mainPeakingFilter	*integer*	Peaking filter main settings.
[osPdForced] **	*boolean*	Whether the OS CDR is forced. **Since: **currentRelease**
p1SlicerThreshold	*integer*	Vertical threshold voltage of p1 slicer (mV).
[rxDecoderEnable]	*boolean*	Rx decode enable.
rxPamMode	*enum: PAM4_NS* | *PAM4_ES* | *NRZ*	Receiver PAM4 mode.
[rxPpm]	*integer*	Frequency offset of local reference clock with respect to the frequency of received data in ppm.
[txPpm]	*integer*	Frequency offset of local reference clock with respect to the transmit data in ppm.
[txPrecoderEnable]	*boolean*	Tx precoder enable.
[txTaps]	*table(numeric string) = integer*	Serdes lane TX pre-emphasis taps.
[vga]	*integer*	Variable gain amplifier value.

2.449.1.17.1.3 BlackhawkPhySerdesLaneModel

brPdEnable	*enum: BR* | *OS*	Baud rate phase detector enable.
[brPdForced] **	*boolean*	Whether the BR CDR is forced. **Since: **currentRelease**
[dfeTaps]	*table(numeric string) = integer*	Serdes lane Decision Feedback Equalizer (DFE) taps.
[eyeLeft]	*integer*	Horizontal left eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
[eyeLower]	*integer*	Vertical lower eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
[eyeRight]	*integer*	Horizontal right eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
[eyeUpper]	*integer*	Vertical upper eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
highFrequencyPeakingFilter	*integer*	High frequency peaking filter control.
linkTrainingEnable	*boolean*	Tx link training enable.
linkTrainingStatus	*enum: link-ready* | *invalid* | *in-training* | *init* | *complete* | *train-remote* | *failed* | *train-local*	Link training status.

lowFrequncyPeakingFilter	*integer*	Low frequency peaking filter control.
m1SlicerThreshold	*integer*	Vertical threshold voltage of m1 slicer (mV), 'channel loss hint' in PAM4 mode should always be 0.
mainPeakingFilter	*integer*	Peaking filter main settings.
[osPdForced] **	*boolean*	Whether the OS CDR is forced. **Since: **currentRelease**
p1SlicerThreshold	*integer*	Vertical threshold voltage of p1 slicer (mV).
rxDecoderEnable	*boolean*	Rx decode enable.
rxPamMode	*enum: PAM4_NS | PAM4_ES | NRZ*	Receiver PAM4 mode.
[rxPpm]	*integer*	Frequency offset of local reference clock with respect to the frequency of received data in ppm.
[txPpm]	*integer*	Frequency offset of local reference clock with respect to the transmit data in ppm.
txPrecoderEnable	*boolean*	Tx precoder enable.
[txTaps]	*table(numeric string) = integer*	Serdes lane TX pre-emphasis taps.
[vga]	*integer*	Variable gain amplifier value.

2.449.1.18 Bcm87728PhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes **	*BlackhawkLtOptionalPhySerdesModel* see 2.449.1.17.1	The SerDes statistics. **Since: 4.30.1
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.19 Bcm87728SystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.

[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes **	*BlackhawkLtOptionalPhySerdesModel* see 2.449.1.17.1	The SerDes statistics. **Since: 4.30.1
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.20 BfnPhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[serdesStats]	*BfnPhySerdesModel* see 2.449.1.20.1	BfnPhy Serdes Statistics.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.20.1 BfnPhySerdesModel

serdes	*table(numeric string) = BfnPhySerdesLaneModel* see 2.449.1.20.1.1	The serdes stats.

2.449.1.20.1.1 BfnPhySerdesLaneModel

ctleDcBoost	*integer*	The DC boost of the CTLE processor.
ctlePfHfGain	*integer*	The peaking filter high frequency gain.
ctlePfLfGain	*integer*	The peaking filter low frequency gain.
ctlePfPeakingFreq	*integer*	The peaking filter center frequency.
dfeDone	*boolean*	Indication of whether DFE is finished.
dfeEye	*integer*	The eye value used by the DFE.
dfeStatus	*integer*	The serdes lane DFE status.
dfeTap	*table(numeric string) = integer*	The serdes lane DFE taps.
eyeMean	*integer*	The mean value of the eye height.

eyeOk	*boolean*	Indication that the eye is acceptable.
eyeWidth	*integer*	The width of the eye.
lane	*integer*	The serdes lane.
mainTap	*integer*	The serdes pre-emphasis main cursor.
postTap	*integer*	The serdes pre-emphasis post cursor.
preTap	*integer*	The serdes pre-emphasis pre cursor.
quickEye	*integer*	The serdes lane eye metric.
ring	*integer*	The serdes ring.
serdesNum	*integer*	The serdes number.
slew	*integer*	The serdes slew value.

2.449.1.21 BfnPhyNiIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes	*table(numeric string) = CredoPhySerdes* see 2.449.1.6.2	A mapping of SerDes lane to SerDes diagnostic information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.21.1 PhyState

changes	*integer*	Number of PHY state changes.
lastChange	*floating point number*	Last time PHY state changed.

value	*enum: errDisabled | unknown | linkUp | linkDown | detectingXcvr | init | autonegotiating*	Current PHY state.

2.449.1.22 BlackhawkPhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes	*BlackhawkPhySerdes* see 2.449.1.22.1	The serdes stats.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.22.1 BlackhawkPhySerdes

serdesStats	*table(numeric string) = BlackhawkPhySerdesLaneModelBase* see 2.449.1.17.1.1	The serdes statistics.
sortedSerdesStats	*list of integer*	Sorted Serdes statistics.

2.449.1.23 CortinaHostPhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
stats	*CortinaXGPCSStatsModel* see 2.449.1.23.1	Additional PHY statistics.
tuningValues **	*TXEqualization* see 2.449.1.23.2	Tuning values. **Since: 4.28.1

2.449.1.23.1 CortinaXGPCSStatsModel

pcsRXErr	*PCSRXErrorCount* see 2.449.1.2.2.12	PCS RX Error Count.
pcsRXGood	*PCSRXGoodCount* see 2.449.1.2.2.13	PCS RX Good Count.
pcsTXErr	*PCSTXErrorCount* see 2.449.1.2.2.14	PCS TX Error Count.
pcsTXGood	*PCSTXGoodCount* see 2.449.1.2.2.15	PCS TX Good Count.

2.449.1.23.2 TXEqualization

main	*integer*	Main tuning value.
post	*integer*	Post tuning value.

2.449.1.24 CortinaLinePhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.

[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
stats	*CortinaXGPCSStatsModel* see 2.449.1.23.1	Additional PHY statistics.

2.449.1.25 CrosspointPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
chipId	*integer*	Chip Id.
chipName	*string*	Chip Name.
coreId	*integer*	Core Id.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
egressSerdesMappings	*list of SerdesMapping* see 2.449.1.25.3	Egress SerDes Mappings.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
ingressSerdesMappings	*list of SerdesMapping* see 2.449.1.25.3	Ingress SerDes Mappings.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
polarities	*list of Polarity* see 2.449.1.25.1	Polarity Values.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
rxEqualizations	*list of RXEqualization* see 2.449.1.25.2	Rx Equalization Values.
[stale]	*boolean*	Status of this phy might be stale.
txEqualizations	*list of TXEqualization* see 2.449.1.25.4	Tx Equalization Values.

2.449.1.25.1 Polarity

lane	*integer*	Lane Id.
rxPolarity	*string*	Rx Polarity Value.
rxPolarityLastChange	*floating point number*	Last change timestamp for Rx Polarity.
txPolarity	*string*	Tx Polarity Value.
txPolarityLastChange	*floating point number*	Last change timestamp for Tx Polarity.

2.449.1.25.2 RXEqualization

afe	*integer*	Rx Afe Equlization Value.
lane	*integer*	Lane Id.

2.449.1.25.3 SerdesMapping

destination	*integer*	Destination Lane Id.
[source]	*integer*	Source Lane Id.

2.449.1.25.4 TXEqualization

lane	*integer*	Lane Id.
[main]	*integer*	Main tuning value.
[post]	*integer*	Post tuning value.
[pre]	*integer*	Pre tuning value.

2.449.1.26 DenaliLinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
coherentPma	*table(numeric string) = Pma* see 2.449.1.2.25	Coherent PMA information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.27 DenaliSystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.

[stale]	*boolean*	Status of this phy might be stale.

2.449.1.28 EnigmaLinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.29 EnigmaSystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.30 EvoraLinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.

[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.31 EvoraSystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.32 FalconPhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.

[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes	*FalconPhySerdes* see 2.449.1.32.1	The Serdes stats.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.32.1 FalconPhySerdes

serdesStats	*table(numeric string) = FalconPhySerdesLaneModel* see 2.449.1.32.1.1	A mapping of SerDes lane to SerDes statistics.

2.449.1.32.1.1 FalconPhySerdesLaneModel

[dfeTaps]	*table(numeric string) = integer*	Serdes lane Decision Feedback Equalizer (DFE) taps.
eyeLeft	*integer*	Horizontal left eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
eyeLower	*integer*	Vertical lower eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
eyeRight	*integer*	Horizontal right eye margin @ 1e-5 as seen by internal diagnostic slicer in mUI.
eyeUpper	*integer*	Vertical upper eye margin @ 1e-5 as seen by internal diagnostic slicer in mV.
lowFrequncyPeakingFilter	*integer*	Low frequency peaking filter control.
mainPeakingFilter	*integer*	Peaking filter main settings.
[rxPpm]	*integer*	Frequency offset of local reference clock with respect to the frequency of received data in ppm.
[txPpm]	*integer*	Frequency offset of local reference clock with respect to the transmit data in ppm.
[txTaps]	*table(numeric string) = integer*	Serdes lane TX pre-emphasis taps.
[vga]	*integer*	Variable gain amplifier value.

2.449.1.33 InternalGen2PhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.

operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.34 InternalGen3PhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.35 SfePhyNacIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdes	*table(numeric string) = IntelNacPhySerdes* see 2.449.1.35.1	A mapping of SerDes lane to SerDes diagnostic information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.35.1 IntelNacPhySerdes

[agc]	*list of integer*	SerDes lane receivier Automatic Gain Coefficients.
[ctleAcGain]	*integer*	SerDes lane Continuous Time Linear Equalizer high frequency gain.
[ctleDcGain]	*integer*	SerDes lane Continuous Time Linear Equalizer low frequency gain.
[dfeEvenTaps]	*list of integer*	Receiver Decision Feedback Equalizer even Coefficients.
[dfeOddTaps]	*list of integer*	Receiver Decision Feedback Equalizer odd Coefficients.
[eyeHeight]	*integer*	Eye height margin in mV.

[rxPpm]	*integer*	Frequency offset of received data with respect to the frequency of local reference clock in parts per million of Hz.
[tdfeEvenTaps]	*list of integer*	Receiver Tail Decision Feedback Equalizer even Coefficients.
[tdfeOddTaps]	*list of integer*	Receiver Tail Decision Feedback Equalizer odd Coefficients.
[tdfeTaps]	*list of integer*	Receiver Tail Decision Feedback Equalizer floating tap positions.
[txTaps]	*IntelNacPhyTxTaps* see 2.449.1.35.1.1	SerDes lane TX pre-emphasis taps.

2.449.1.35.1.1 IntelNacPhyTxTaps

main	*integer*	The main cursor tap.
post1	*integer*	The post cursor tap.
pre1	*integer*	The pre cursor tap.

2.449.1.36 SfePhyNimIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.37 SwitchAsic400gPhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.
operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.38 InternalPhyIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[interruptCount]	*integer*	PHY interrupt counter.
[laneCount]	*enum: 4 | 1 | 2 | 10 | 8*	PHY lane count to support operational speed.
[modulation]	*enum: PAM-4 | NRZ*	PHY modulation configured on the interface.

operSpeed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
phyState	*PhyState* see 2.449.1.21.1	PHY state information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.39 MillenioLinePhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.40 MillenioSystemPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.

[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.41 PhyDs280ChipStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.42 PhyDs280IntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdesStats	*table(numeric string) = Ds280PhySerdes* see 2.449.1.42.1	A mapping of SerDes lane to SerDes diagnostic information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.42.1 Ds280PhySerdes

[dfeTaps]	*list of floating point number*	SerDes lane Decision Feedback Equalizer (DFE) taps.
[horizontalEye]	*floating point number*	Eye margin in UI.
[txTaps]	*Ds280PhyTxTaps* see 2.449.1.42.1.1	SerDes lane TX pre-emphasis taps.

[verticalEye]	*floating point number*	Eye margin in mV.

2.449.1.42.1.1 Ds280PhyTxTaps

main	*integer*	The main cursor tap.
post	*integer*	The post cursor tap.
pre	*integer*	The pre cursor tap.

2.449.1.43 PhyDs280LineIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdesStats	*table(numeric string) = Ds280PhySerdes* see 2.449.1.42.1	A mapping of SerDes lane to SerDes diagnostic information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.44 PhyDs280SystemIntfStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
serdesStats	*table(numeric string) = Ds280PhySerdes* see 2.449.1.42.1	A mapping of SerDes lane to SerDes diagnostic information.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.45 RawPhyStatus

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
text	*string*	The text from CLI output.

2.449.1.46 ShortfinLinePhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
mdiStatus **	*MdiCrossoverModel* see 2.449.1.14.1	MDI/MDI-X status. **Since: 4.30.0
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
speedDownshiftedStatus	*SpeedDownshiftedModel* see 2.449.1.46.1	Speed downshifted status.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.46.1 SpeedDownshiftedModel

changes	*integer*	Number of speed downshift changes.
lastChange	*floating point number*	Last time speed downshift changed.
state	*boolean*	Speed downshifted for mGig PHYs due to unstable link.

2.449.1.47 ShortfinPhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
chipStatus	*PhyChipStatus* see 2.449.1.2.21	Shortfin chip status.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
mdioCommand	*integer*	MDIO Command Handler command selection.
mdioData1	*integer*	MDIO Command Handler parameter 1 for command.
mdioData2	*integer*	MDIO Command Handler parameter 2 for command.
mdioData3	*integer*	MDIO Command Handler parameter 3 for command.
mdioData4	*integer*	MDIO Command Handler parameter 4 for command.
mdioData5	*integer*	MDIO Command Handler parameter 5 for command.
mdioStatus	*integer*	MDIO Command Handler command status.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.
[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
speed	*enum: 42Gbps | 53Gbps | 200Gbps | 1Gbps | 100Mbps | 50Gbps | 25Gbps | 10Mbps | 40Gbps | unknown | 27Gbps | 10Gbps | 100Gbps | 11Gbps | 106Gbps | 21Gbps | 2.5Gbps | 400Gbps | 5Gbps*	Operational speed of the interface.
[stale]	*boolean*	Status of this phy might be stale.

2.449.1.48 ShortfinSystemPhyModel

[chip]	*PhyChipStatus* see 2.449.1.2.21	Phy chip information.
description	*PhyDescription* see 2.449.1.2.24	Phy description.
[fec]	*Fec* see 2.449.1.2.1	FEC information.
[pcs]	*Pcs* see 2.449.1.2.8	PCS information.

[pma]	*Pma* see 2.449.1.2.25	PMA information.
[preFecBer]	*PreFecBitError* see 2.449.1.2.30.12	Pre FEC bit error rate of the link.
[rs]	*Rs* see 2.449.1.2.38	Reconciliation sublayer information.
[stale]	*boolean*	Status of this phy might be stale.
[usxgmiiCtrl1]	*integer*	USXGMIIM_USXGMII_CTRL1 - USXGMII Control 1.
[usxgmiiErrStatus]	*integer*	USXGMIIM_ERROR_STATUS - ERROR Status.
[usxgmiiRxAutoneg]	*integer*	USXGMIIM_AN_RX_MESSAGE - Autoneg RX Message.
[usxgmiiStatus0]	*integer*	USXGMIIM_USXGMII_STATUS0 - USXGMII Status0.
[usxgmiiStatus1]	*integer*	USXGMIIM_USXGMII_STATUS1 - USXGMII Status1.
[usxgmiiTxConfig]	*integer*	USXGMIIM_TX_CONFIG_REG - TX Config Register.

2.449.1.49 Transceiver

mediaType	*TransceiverMediaType* see 2.449.1.49.1	Transceiver media type.
[vendorSn]	*string*	Vendor serial number of transceiver.

2.449.1.49.1 TransceiverMediaType

changes	*integer*	Number of changes in transceiver presence.
lastChange	*floating point number*	Last time transceiver presence changed.
value	*string*	Transceiver media type.

2.450 show interfaces (phy diag error-correction histogram) | (phy diag error-correction histogram (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 2

interfaces	*table(Interface name) = IntfFecHistogramPhy* see 2.450.1	Map of interface to FEC histogram.

2.450.1 IntfFecHistogramPhy

phys	*list of FecHistogram* see 2.450.1.1	Fec histogram for all phys in order of topology.

2.450.1.1 FecHistogram

Could also be an instance of:
B52LineFecHistogram 2.450.1.2
B52SystemFecHistogram 2.450.1.3
BabbageLineFecHistogram 2.450.1.4
BabbageSystemFecHistogram 2.450.1.5
MillenioLineFecHistogram 2.450.1.6
MillenioSystemFecHistogram 2.450.1.7

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.2 B52LineFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.3 B52SystemFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.4 BabbageLineFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.5 BabbageSystemFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.6 MillenioLineFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.
[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.450.1.7 MillenioSystemFecHistogram

[correctedCodewords]	*table(string) = FecHistoBin* see 2.449.1.2.2.9	The number of codewords indexed by the number or range of symbol errors in the codeword.
[description]	*PhyDescription* see 2.449.1.2.24	Phy description.
[sampled]	*boolean*	Corrected codewords counts are sampled.

[uncorrectedCodewords]	*FecHistoBin* see 2.449.1.2.2.9	The number of codewords with uncorrected errors.

2.451 show interfaces (phy diag test pattern) | (phy diag test pattern (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = IntfTestPatternStatus* see 2.451.1	Mapping between an interface and test pattern config.

2.451.1 IntfTestPatternStatus

capabilities	*string*	Supported test-patterns.
[rx]	*TestPatternMode* see 2.451.1.1	Receive test pattern mode.
[tx]	*TestPatternMode* see 2.451.1.1	Transmit test pattern mode.

2.451.1.1 TestPatternMode

[configured]	*enum: patternNone | prbs9 | prbs31 | prbs49 | prbs23 | prbs63 | prbs7 | prbs13 | prbs58 | prbs15 | prbs11*	Configured test pattern mode.
[operational]	*enum: patternNone | prbs9 | prbs31 | prbs49 | prbs23 | prbs63 | prbs7 | prbs13 | prbs58 | prbs15 | prbs11*	Operational test pattern mode.

2.452 show interfaces (phy diag test pattern counters) | (phy diag test pattern counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = IntfTestPatternCounters* see 2.452.1	Mapping of interface to test pattern counter data.

2.452.1 IntfTestPatternCounters

lanes	*table(numeric string) = LaneTestPatternCounters* see 2.452.1.1	Mapping of lanes that are a part of the interface to test pattern counter data.

2.452.1.1 LaneTestPatternCounters

Could also be an instance of:
LaneTestPatternDetail 2.452.1.2

bitErrors	*integer*	Total bit errors.
burstCount	*integer*	Number of bursts of bit errors.
largestBurst	*integer*	Largest burst of bit errors.
lastErrorTime	*floating point number*	Last time bit errors were observed.
linkState	*enum: locked | notLocked | notApplicable*	Lock status of received diagnostic test pattern.

2.452.1.2 LaneTestPatternDetail

berSinceLastLock	*floating point number*	Bit error rate since last lock.
bitErrorRate	*floating point number*	Bit error rate since last clear.
bitErrors	*integer*	Total bit errors.
bitErrorsSinceLastLock	*integer*	Total bit errors since last lock.
bitRate	*floating point number*	Bit rate of the lane in Gbps.
burstCount	*integer*	Number of bursts of bit errors.
largestBurst	*integer*	Largest burst of bit errors.
lastErrorTime	*floating point number*	Last time bit errors were observed.
lastLinkStateChange	*floating point number*	Time of last link state change.
linkState	*enum: locked | notLocked | notApplicable*	Lock status of received diagnostic test pattern.
linkStateChanges	*integer*	Number of link state changes since last clear.
lockLostLatchState **	*enum: waitingForLock | lost | notLost | unknown*	Status of the lock latch since RX became operational. Reset when a new RX pattern becomes operational. **Since: 4.23.2
totalBits	*floating point number*	Total bits received since last clear in Gb.
totalBitsSinceLastLock	*floating point number*	Total bits received since last lock in Gb.

2.453 show interfaces (phy diag test pattern counters detail) | (phy diag test pattern counters detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

`interfaces`	*table(Interface name)* = *IntfTestPatternDetail* see 2.453.1	Mapping of interface to detailed test pattern data.

2.453.1 IntfTestPatternDetail

`lanes`	*table(numeric string)* = *LaneTestPatternDetail* see 2.452.1.2	Mapping of lanes to detailed test pattern status.
`lastClear`	*floating point number*	The last time the test pattern counterswere cleared.
`[operationalTestPattern]`	*enum: patternNone* | *prbs9* | *prbs31* | *prbs49* | *prbs23* | *prbs63* | *prbs7* | *prbs13* | *prbs58* | *prbs15* | *prbs11*	Operational test pattern.

2.454 show interfaces (phy modulation) | (phy modulation (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switch-card MODSLOT_SWITCHCARD)))) [...]

Could also be an instance of:
InterfacesPhyModulation 2.454.1.1
InterfacesPhyLineRate 2.454

interfaceLineRate	*table(Interface name) = InterfacePhyLineRate* see 2.454.1	Mapping between an interface and phy line rate information.

2.454.1 InterfacePhyLineRate

availableLineRates	*InterfaceAvailableLine-Rates* see 2.454.1.1	Line rate choices available for configuration on this interface.
lineRateConfigured	*enum: 100 | 400 | 200 | none | 300*	Configured line rate. 'none' implies that no configuration has been applied on this interface.
lineRateStatus	*enum: 100 | 400 | 200 | none | 300*	Line rate in operation. 'none' implies that this interface is inactive.
modulationStatus	*enum: qam16 | none | qam8 | dpQpsk*	Modulation in operation. 'none' implies that this interface is inactive.

2.454.1.1 InterfaceAvailableLineRates

[lr100g]	*boolean*	100g line rate.
[lr200g]	*boolean*	200g line rate.
[lr300g]	*boolean*	300g line rate.
[lr400g]	*boolean*	400g line rate.
interfaceModulation	*table(Interface name) = InterfacePhyModulation* see 2.454.2	Mapping between an interface and phy modulation information.

2.454.2 InterfacePhyModulation

availableModulations	*InterfaceAvailable-Modulations* see 2.454.2.1	Modulation choices available for configuration on this interface.
modulationConfigured	*enum: qam16 | none | qam8 | dpQpsk*	Configured modulation. 'none' implies that no configuration has been applied on this interface.
modulationStatus	*enum: qam16 | none | qam8 | dpQpsk*	Modulation in operation. 'none' implies that this interface is inactive.

2.454.2.1 InterfaceAvailableModulations

[dpQpsk]	*boolean*	dual-polarization quadrature phase-shift keying.
[qam16]	*boolean*	16 state quadrature amplitude modulation.

[qam8]	*boolean*	8 state quadrature amplitude modulation.

2.455 show interfaces (phy precoding) | (phy precoding (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfacePrecoding* see 2.455.1	Mapping between an interface and phy precoding information.

2.455.1 InterfacePrecoding

lanes	*table(numeric string) = PrecodingLaneDetail* see 2.455.1.1	Detailed precoding information on per-lane basis.
rxConfig	*enum: default | off | on*	Configured precoding on receiver.
txConfig	*enum: default | off | on*	Configured precoding on transmitter.

2.455.1.1 PrecodingLaneDetail

requestStatus	*PrecodingRequestStatus* see 2.455.1.1.1	Precoding request status on per-lane basis.
txRxStatus	*PrecodingTxRxStatus* see 2.455.1.1.2	Precoding status of Tx and Rx on per-lane basis.

2.455.1.1.1 PrecodingRequestStatus

requestedByPeer	*boolean*	Link partner requests precoding.
requestedOfPeer	*boolean*	Local device requests precoding.

2.455.1.1.2 PrecodingTxRxStatus

rxEnabled	*boolean*	Precoding is enabled on receiver.
txEnabled	*boolean*	Precoding is enabled on transmitter.

2.456 show interfaces (phy skew) | (phy skew (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfacesPhySkew* see 2.456.1	Active interface skew values.

2.456.1 InterfacesPhySkew

receiver	*SkewValues* see 2.456.1.1	Receiver skew values.
transmitter	*SkewValues* see 2.456.1.1	Transmitter skew values.

2.456.1.1 SkewValues

[calibrationFailureReason]	*enum: linkFlap | concurrentOperation | linkDown | modulation | uncorrectableCodeBlocks-Increment*	Reason for failed calibration.
[calibrationStatus]	*enum: ok | failed*	Status of skew calibration.
skewH	*floating point number*	Horizontal Polarization Skew (picoseconds).
skewV	*floating point number*	Vertical Polarization Skew (picoseconds).
source	*enum: default | computing | persistent*	Source of the active skew values.

2.457 show interfaces (phy skew persistent) | (phy skew persistent (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switch-card MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfacesPhySkewPstore* see 2.457.1	Active interface skew values.

2.457.1 InterfacesPhySkewPstore

[receiver]	*PstoreSkewValues* see 2.457.1.1	Receiver skew values.
[transmitter]	*PstoreSkewValues* see 2.457.1.1	Transmitter skew values.
version	*integer*	Skew persistent store version.
xcvrSn	*string*	Transceiver serial number.

2.457.1.1 PstoreSkewValues

skewH	*floating point number*	Horizontal Polarization Skew (picoseconds).
skewV	*floating point number*	Vertical Polarization Skew (picoseconds).
state	*enum: inactive | active*	Indicates if the values read from non-volatile memory are actively used or not.

2.458 show interfaces (priority-flow-control) | (priority-flow-control (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 3

interfaceCounters	*table(Interface name)* = *PfcFrameCountersModel* see 2.458.1	A mapping of interface name to priority flow control counters.
interfaceStatuses	*table(Interface name)* = *PfcIntfStatusModel* see 2.458.2	A mapping of interface name to priority flow control status.
status	*PfcStatusModel* see 2.458.3	Status of priority flow control at a global level.
[watchdogStatus]	*PfcWatchdogStatusModel* see 2.458.4	Status of priority flow control watchdog at a global level.

2.458.1 PfcFrameCountersModel

[rxFrames]	*integer*	Number of priority flow control frames received.
[txFrames]	*integer*	Number of priority flow control frames transmitted.

2.458.2 PfcIntfStatusModel

active	*boolean*	Indicates whether priority flow control is operationally active.
enabled	*boolean*	Indicates whether priority flow control is enabled.
[note]	*string*	Informational note on the status of priority flow control.
priorities	*list of integer*	Priorities for which priority flow control is enabled.
[watchdogAction]	*enum: drop | errdisable | notify-only*	Action taken by the priority flow control watchdog.
watchdogActive **	*boolean*	Indicates whether priority flow control watchdog is operationally active. **Since: 4.27.2
watchdogEnabled	*boolean*	Indicates whether priority flow control watchdog is enabled.
[watchdogTimer]	*PfcWatchdogTimerModel* see 2.458.2.1	Timer attributes of the priority flow control watchdog.

2.458.2.1 PfcWatchdogTimerModel

forcedRecovery	*boolean*	Indicates whether forced recovery has been configured for the priority flow control watchdog.
[operPollInterval]	*floating point number*	Polling interval (in seconds) actually operational for the priority flow control watchdog.
pollInterval	*floating point number*	Polling interval (in seconds) configured for the priority flow control watchdog.
recoveryTime	*floating point number*	Wait time (in seconds) after which the priority flow control watchdog marks a transmit queue as recovered.
timeout	*floating point number*	Time (in seconds) for the priority flow control watchdog to expire.

2.458.3 PfcStatusModel

enabled	*boolean*	Indicates whether priority flow control is enabled at a global level.
[priorities]	*list of integer*	Priority values for which priority flow control is supported by the hardware.
rxPfcPrioritiesHonored **	*list of integer*	Priorities for which PFC pause framesare honored. **Since: 4.22.0
supported	*boolean*	Indicates whether priority flow control is supported by the hardware.

2.458.4 PfcWatchdogStatusModel

[action]	*enum: drop* | *errdisable* | *notify-only*	Action taken by the priority flow control watchdog.
[forcedNd] **	*boolean*	Indicates whether port action non-disruptive-only behaviour is on or off. **Since: 4.21.3
[hwMonitoredPri] **	*list of integer*	Priority values configured as Hardware monitored priorities. **Since: 4.23.2
[nDAction] **	*enum: invalid* | *forward*	Action taken on non-disruptive queues by the priority flow control watchdog. **Since: 4.22.0
[nDPriorities]	*PfcWatchdogNDPriorities-Model* see 2.458.4.1	Non-disruptive priorities supported by the priority flow control watchdog.
[overrideAction] **	*boolean*	Indicates whether override action has been enabled to unconditionally drop packets. **Since: 4.21.3
supported	*boolean*	Indicates whether priority flow control watchdog is supported at a global level.
[timer]	*PfcWatchdogTimerModel* see 2.458.2.1	Timer attributes of the priority flow control watchdog.

2.458.4.1 PfcWatchdogNDPrioritiesModel

[priorities]	*list of integer*	Priority values configured as non-disruptive for the priority flow control watchdog.
supported	*boolean*	Indicates whether non-disruptive priorities are supported by the priority flow control watchdog.

2.459 show interfaces (priority-flow-control counters) | (priority-flow-control counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 3

interfaceCounters	*table(Interface name)* = *PfcFrameCountersModel* see 2.458.1	A mapping of interface name to priority flow control counters.

2.460 show interfaces (priority-flow-control counters detail) | (priority-flow-control counters detail (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 3

interfaces	*table(Interface name)* = *PfcIntfDetailedCounters-Model* see 2.460.1	A mapping of interface name to priority flow control detailed counters.

2.460.1 PfcIntfDetailedCountersModel

priorities	*table(numeric string)* = *PfcFrameCountersModel* see 2.458.1	A mapping of priority value to priority flow control counters.

2.461 show interfaces (priority-flow-control counters watchdog) | (priority-flow-control counters watchdog (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 2

interfaces	*table(Interface name) = PfcWatchdogIntfCounters-Model* see 2.461.1	A mapping of interface name to priority flow control watchdog counters.

2.461.1 PfcWatchdogIntfCountersModel

txQueues	*table(numeric string) = PfcWatchdogTxQueue-CountersModel* see 2.461.1.1	A mapping of transmit queue id to priority flow control watchdog counters.

2.461.1.1 PfcWatchdogTxQueueCountersModel

lastRecoveryTimestamp **	*floating point number*	Timestamp of last recorded recovery event for transmit queue. **Since: 4.28.1
lastStuckTimestamp **	*floating point number*	Timestamp of last recorded stuck event for transmit queue. **Since: 4.28.1
queueType	*enum: unicast | multicast*	Type of the transmit queue.
recoveryCount	*integer*	Number of times the transmit queue recovered.
stuckCount	*integer*	Number of times the transmit queue was stuck.

2.462 show interfaces (priority-flow-control counters watchdog drop) | (priority-flow-control counters watchdog drop (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 3

interfaces	*table(Interface name) = PfcWatchdogIntfEgress-DropCountersModel* see 2.462.1	A mapping of interface name to priority flow control watchdog drop counters.

2.462.1 PfcWatchdogIntfEgressDropCountersModel

egressCounters	*table(numeric string) = PfcWatchdogEgressDrop-CountersModel* see 2.462.1.1	A mapping of transmit queue id to priority flow control watchdog drop counters.
ingressCounters	*table(numeric string) = PfcWatchdogIngressDrop-CountersModel* see 2.462.1.2	A mapping of traffic class to priority flow control watchdog ingress drop counters.

2.462.1.1 PfcWatchdogEgressDropCountersModel

[outAggregateByteDrops]	*integer*	Bytes dropped at the transmit queue.
[outAggregateDrops]	*integer*	Packets dropped at the transmit queue.
[outCurrentByteDrops]	*integer*	Bytes dropped in current stuck-recovery cycle.
[outCurrentDrops]	*integer*	Packets dropped in current stuck-recovery cycle.
[queueType]	*enum: unicast | multicast*	Type of the transmit queue.

2.462.1.2 PfcWatchdogIngressDropCountersModel

[aggregateByteDrops]	*integer*	Bytes dropped on the ingress side.
[aggregateDrops]	*integer*	Packets dropped on the ingress side.
[currentByteDrops]	*integer*	Bytes dropped in current stuck-recovery cycle.
[currentDrops]	*integer*	Number of packets dropped in current stuck-recovery cycle on the ingress side.

2.463 show interfaces (priority-flow-control status) | (priority-flow-control status (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

revision 2

interfaceStatuses	*table(Interface name) = PfcIntfStatusModel* see 2.458.2	A mapping of interface name to priority flow control status.
status	*PfcStatusModel* see 2.458.3	Status of priority flow control at a global level.
[watchdogStatus]	*PfcWatchdogStatusModel* see 2.458.4	Status of priority flow control watchdog at a global level.

2.464 show interfaces (status errdisabled) | (status errdisabled (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaceStatuses	*table(Interface name) = InterfaceCauseModel* see 2.464.1	Map interfaces to errdisabled status.

2.464.1 InterfaceCauseModel

[causes]	*list of string*	errdisabled causes for an interface.
[description]	*string*	Interface description.

[status]	*string*	Interface enabled state reason.

2.465 show interfaces (tap | tool) | (tap | tool (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

tapProperties	*table(Interface name) = TapProperty* see 2.465.1	Dictionary of tap port properties keyed by interface name.
toolProperties	*table(Interface name) = ToolProperty* see 2.465.2	Dictionary of tool port properties keyed by interface name.

2.465.1 TapProperty

allowedVlans	*string*	The set of allowed vlans on a tap port in the range 1-4094.
configuredMode	*enum: dot1qTunnel | tap-tool | tap | routed | trunk | tool | access*	The configured switchport mode of an interface.
[details]	*Details* see 2.465.1.1	Detailed tap properties.
innerPortIdentity **	*integer*	The inner port identity of a tap port. **Since: 4.20.5
linkStatus	*enum: signal | inactive | non-tap | unknown | maint-down | errdisabled | non-tool | disabled | admin | tap | notapplicable | notconnect | noSignal | tool | initializing | uninitialized*	The link status of an interface.
nativeVlan	*integer*	The native vlan associated with a tap port.
portIdentity	*integer*	The port identity of a tap port.
truncationSize	*integer*	The size to which packet is to be truncated, in bytes.
[tunnel] **	*Tunnel* see 2.465.1.2	Tunnel tap actions. **Since: 4.25.2

2.465.1.1 Details

aclsAppliedPerType	*table(string) = AclsAppliedPerType* see 2.465.1.1.1	The list of ACLs applied on a port keyed by ACL type.
defaultGroup	*string*	The default group associated with a tap port.
destMac **	*MAC address*	Destination MAC address. **Since: 4.28.0
[groups]	*list of string*	List of default groups for the tap port.
mplsPop **	*boolean*	Indicates whether MPLS pop is enabled on a tap port. **Since: 4.20.5
srcMac **	*MAC address*	Source MAC address. **Since: 4.28.0

[toolPorts]	*list of Interface name*	List of tool ports for the tap port.
vxlanStrip **	*boolean*	Indicates whether VXLAN strip is enabled on a tap port. **Since: 4.20.5

2.465.1.1.1 AclsAppliedPerType

aclsApplied	*list of string*	List of ACL names applied on a port.

2.465.1.2 Tunnel

brTagStrip	*boolean*	Indicates whether 802.1BR Tag strip is enabled on a tap port.
mplsPop	*boolean*	Indicates whether MPLS pop is enabled on a tap port.
vnTagStrip	*boolean*	Indicates whether VN Tag strip is enabled on a tap port.
vxlanStrip	*boolean*	Indicates whether VXLAN strip is enabled on a tap port.

2.465.2 ToolProperty

allowedVlans	*string*	The set of allowed vlans on a tool port in the range 1-4094.
configuredMode	*enum: dot1qTunnel | tap-tool | tap | routed | trunk | tool | access*	The configured switchport mode of an interface.
[details]	*Details* see 2.465.2.1	Detailed tool properties.
identificationTag	*enum: dot1q | qinq | Off*	The identity tag requirement for packets recieved on a tool port.
linkStatus	*enum: signal | inactive | non-tap | unknown | maint-down | errdisabled | non-tool | disabled | admin | tap | notapplicable | notconnect | noSignal | tool | initializing | uninitialized*	The link status of an interface.
timestampMode	*enum: None | header | before-fcs | NA | replace-fcs*	The timestamping mode on a packet recieved on a tool port.
[tunnel] **	*Tunnel* see 2.465.2.2	Tunnel tool actions. **Since: 4.25.2

2.465.2.1 Details

[aclsAppliedPerType]	*table(string) = AclsAppliedPerType* see 2.465.1.1.1	The dictionary of ACLs applied on a portkeyed by ACL type.
[groups]	*list of string*	List of groups of the tool port.
[removedVlans]	*string*	The set of 802.1Q tags to remove on a tool port.

truncationSize	*integer*	The size to which packet is to be truncated, in bytes.

2.465.2.2 Tunnel

brTagStrip	*boolean*	Indicates whether 802.1BR Tag strip is enabled on a tool port.
mplsPop	*boolean*	Indicates whether MPLS pop is enabled on a tool port.
vnTagStrip	*boolean*	Indicates whether VN Tag strip is enabled on a tool port.

2.466 show interfaces Fabric priority-flow-control

enabled	*boolean*	Indicates whether priority flow control is enabled.
intfName	*string*	Fabric interface name.
priorities	*list of integer*	Priorities for which priority flow control is enabled.
status	*PfcStatusModel* see 2.458.3	Status of priority flow control at a global level.
[watchdogStatus]	*PfcWatchdogStatusModel* see 2.458.4	Status of priority flow control watchdog at a global level.

2.467 show interfaces [...]

interfaces	*table(Interface name)* = *InterfaceStatus* see 2.467.1	Interface Status.

2.467.1 InterfaceStatus

Could also be an instance of:
InterfaceConfigSanity 2.467.7
EthInterfaceStatus 2.467.3
EthPhyInterfaceStatus 2.467.4
VlanInterfaceStatus 2.467.10
LagEthInterfaceStatus 2.467.6
InternalRecircInterfaceStatus 2.467.5
VxlanInterfaceStatus 2.467.11
ApplicationInterfaceStatus 2.467.2
TunnelIntfConfigSanity 2.467.8
TunnelIntfStatus 2.467.9

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.

[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.1.1 InterfaceAddressIp4Base

Could also be an instance of:
InterfaceAddress 2.467.1.2
InterfaceAddressBrief 2.467.1.3

2.467.1.2 InterfaceAddress

broadcastAddress	*string*	IP used for broadcast (typically '255.255.255.255').
[dhcp] **	*boolean*	Whether the address of this interface is determined via DHCP. **Since: 4.18.0
primaryIp	*IpAddress* see 2.467.1.2.1	Primary IP address.
secondaryIps	*table(string) = IpAddress* see 2.467.1.2.1	Secondary IP addresses.
[secondaryIpsOrderedList]	*list of IpAddress* see 2.467.1.2.1	Secondary IP addresses ordered list.
[unnumberedIntf] **	*interface name*	Donor interface. **Since: 4.22.1
virtualIp	*IpAddress* see 2.467.1.2.1	Virtual IP address.
virtualSecondaryIps **	*table(string) = IpAddress* see 2.467.1.2.1	Virtual Secondary IP addresses. **Since: 4.18.0

virtualSecondaryIpsOrdered-List **	*list of IpAddress* see 2.467.1.2.1	Virtual Secondary IP addresses ordered list. **Since: 4.18.0

2.467.1.2.1 IpAddress

address	*IPv4 address*	IPv4 address of this interface.
maskLen	*integer*	IP address mask length (e.g 24 for a /24).

2.467.1.3 InterfaceAddressBrief

ipAddr	*IpAddress* see 2.467.1.2.1	Interface IP address.
[unnumberedIntf]	*interface name*	Donor interface.

2.467.1.4 InterfaceAddressIp6Base

Could also be an instance of:
InterfaceAddressIp6 2.467.1.5

2.467.1.5 InterfaceAddressIp6

[addrSource] **	*enum: slaac | manual*	How the address is configured on the interface. **Since: 4.20.0
globalAddressesAreVirtual **	*boolean*	Whether configured addresses are physical or virtual. **Since: 4.24.2
globalUnicastIp6s	*list of IpAddress* see 2.467.1.5.1	Global-unicast IPv6 addresses.
[linkLocalIp6]	*IpAddress* see 2.467.1.5.1	Link-local IPv6 address.

2.467.1.5.1 IpAddress

active	*boolean*	Whether or not this IPv6 address is currently active.
address	*IPv6 address*	An IPv6 address (e.g. 'fc00::1234').
dadfailed	*boolean*	Whether or not this IPv6 address is a duplicate.
leastpref **	*boolean*	This address is least-preferred. **Since: 4.28.2
subnet	*string*	The subnet of this address (e.g. 'fc00::/64').

2.467.2 ApplicationInterfaceStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.

forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
functionInterfaces	*table(Interface name) = FunctionInterfaceSummary* see 2.467.2.1	Functional interfaces running on thisApplication FPGA connector interface.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru	*integer*	L2 MRU (bytes).
l3MtuConfigured	*boolean*	Per-interface MTU is set under interface configuration mode.
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.

mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
owner	*string*	Owner application.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.2.1 FunctionInterfaceSummary

hasRx **	*boolean*	Whether the function intf uses rx. **Since: 4.29.0
hasTx **	*boolean*	Whether the function intf uses tx. **Since: 4.29.0
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
[isSecondaryRx] **	*boolean*	Whether the function intf rx is secondary. **Since: 4.29.0
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
name	*interface name*	Name of the interface.

2.467.3 EthInterfaceStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.

hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
[interfaceCounters]	*EthInterfaceCounters* see 2.467.3.1	Interface Counters and related information.
[interfaceMembership]	*string*	Shows whether the interface is part of a monitor destination or a Port-channel.
[interfaceStatistics]	*EthInterfaceStatistics* see 2.467.3.2	Interface statistics, such as rates.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.3.1 EthInterfaceCounters

[counterRefreshTime]	*floating point number*	UTC of last counter refresh.
inBroadcastPkts	*integer*	Input broadcast packets.
[inDiscards]	*integer*	Input packets with errors.
inMulticastPkts	*integer*	Input multicast packets.
inOctets	*integer*	Input octets.
[inTotalPkts] **	*integer*	Input packets. **Since: 4.20.0

inUcastPkts	*integer*	Input unicast packets.
[inputErrorsDetail]	*PhysicalInputErrors* see 2.467.3.1.1	Details on the types of input errors with physical layer counters.
[lastClear]	*floating point number*	Last clear of interface counters.
[lastUpdateTimestamp] **	*floating point number*	Time of last update. **Since: 4.21.3
linkStatusChanges	*integer*	Number of link status changes since last clear.
outBroadcastPkts	*integer*	Output broadcast packets.
[outDiscards]	*integer*	Output packets with errors.
outMulticastPkts	*integer*	Output multicast packets.
outOctets	*integer*	Output octets.
[outTotalPkts] **	*integer*	Output packets. **Since: 4.28.2
outUcastPkts	*integer*	Output unicast packets.
[outputErrorsDetail]	*PhysicalOutputErrors* see 2.467.3.1.2	Details on the types of output errors with physical layer counters.
totalInErrors	*integer*	Total number of input errors.
totalOutErrors	*integer*	Total number of output errors.

2.467.3.1.1 PhysicalInputErrors

alignmentErrors	*integer*	Alignment errors.
fcsErrors	*integer*	FCS errors.
giantFrames	*integer*	Number of giant frames received.
runtFrames	*integer*	Number of runt frames received.
rxPause	*integer*	Flow control pause frames received.
symbolErrors	*integer*	Symbol errors.

2.467.3.1.2 PhysicalOutputErrors

collisions	*integer*	Sum of single and multiple collisions.
deferredTransmissions	*integer*	Deferred transmissions.
lateCollisions	*integer*	Late collision frames.
txPause	*integer*	Flow control pause frames sent.

2.467.3.2 EthInterfaceStatistics

inBitsRate	*floating point number*	input bits rate.
inPktsRate	*floating point number*	input packets rate.
outBitsRate	*floating point number*	output bits rate.
outPktsRate	*floating point number*	output packets rate.
updateInterval	*floating point number*	Stats update interval in seconds.

2.467.4 EthPhyInterfaceStatus

autoNegotiate	*enum: unknown* | *off* | *success* | *negotiating* | *failed*	Current auto-negotiation state.
bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
duplex	*enum: duplexHalf* | *duplexUnknown* | *duplexFull*	Duplex status of the interface.
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
[interfaceCounters]	*EthInterfaceCounters* see 2.467.3.1	Interface Counters and related information.
[interfaceMembership]	*string*	Shows whether the interface is part of a monitor destination or a Port-channel.
[interfaceStatistics]	*EthInterfaceStatistics* see 2.467.3.2	Interface statistics, such as rates.

interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lanes] **	*integer*	Number of lanes on the interface. **Since: 4.21.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[loopbackMode]	*enum: loopbackPhyRemote* | *loopbackMac* | *loopbackNone* | *loopbackPhy*	Loopback mode of the interface.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.5 InternalRecircInterfaceStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.

forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
[interfaceCounters]	*EthInterfaceCounters* see 2.467.3.1	Interface Counters and related information.
[interfaceMembership]	*string*	Shows whether the interface is part of a monitor destination or a Port-channel.
[interfaceStatistics]	*EthInterfaceStatistics* see 2.467.3.2	Interface statistics, such as rates.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru	*integer*	L2 MRU (bytes).
l3MtuConfigured	*boolean*	Per-interface MTU is set under interface configuration mode.
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.

lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.6 LagEthInterfaceStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
fallbackEnabled	*boolean*	LAG fallback configured.
[fallbackEnabledType]	*enum: fallbackStatic* | *fallbackIndividual* | *fallbackNone*	The type of LAG fallback configured.
[fallbackMode]	*enum: enabledLocally* | *unconfigured* | *configured* | *monitoring* | *passive* | *enabled* | *initialized* | *na*	The status of LAG fallback.
[fallbackTimeout]	*floating point number*	The timeout for fallback to occur.
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.

interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
[interfaceCounters]	*EthInterfaceCounters* see 2.467.3.1	Interface Counters and related information.
[interfaceMembership]	*string*	Shows whether the interface is part of a monitor destination or a Port-channel.
[interfaceStatistics]	*EthInterfaceStatistics* see 2.467.3.2	Interface statistics, such as rates.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
memberInterfaces	*table(string) = LagMemberInformation* see 2.467.6.1	The interfaces that are in this LAG. None if no information is available on the interface.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.6.1 LagMemberInformation

bandwidth	*integer*	Bandwidth (bit).
duplex	*enum: duplexHalf* | *duplexUnknown* | *duplexFull*	Duplex status of the interface.

2.467.7 InterfaceConfigSanity

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.

forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured	*boolean*	Per-interface MTU is set under interface configuration mode.
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.

[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.8 TunnelIntfConfigSanity

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured	*boolean*	Per-interface MTU is set under interface configuration mode.
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.

lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.
tunDecapDipOnlyMatch **	*boolean*	Terminate tunnel from any source. **Since: 4.30.0
tunDiagnosticsInfo	*list of string*	Diagnostic messages.
tunDstAddr	*IP address*	Address for destination.
[tunDstHost] **	*string*	Tunnel destination hostname. **Since: 4.29.1
tunHwFwd	*boolean*	Hardware forwarding.
tunIKey	*integer*	Output key.
tunMode	*enum: tunnelIntfModeIpsec* | *tunnelIntfModeUnspecified* | *tunnelIntfModeGre* | *tunnelIntfModeIpip*	Tunnel mode.
tunMtu	*integer*	Tunnel transport MTU.
tunOKey	*integer*	Input key.
tunPathMtuDiscovery	*boolean*	Tunnel path MTU discovery.
tunSrcAddr	*IP address*	Address for source.
tunTos	*integer*	Tunnel TOS.
tunTtl	*integer*	Tunnel TTL.
tunUnderlayVrf	*string*	Tunnel underlay VRF.

2.467.9 TunnelIntfStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.

forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).

name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.
tunDecapDipOnlyMatch **	*boolean*	Terminate tunnel from any source. **Since: 4.30.0
tunDiagnosticsInfo **	*list of string*	Diagnostic messages. **Since: 4.21.3
tunDstAddr	*IP address*	Address for destination.
[tunDstHost] **	*string*	Tunnel destination hostname. **Since: 4.29.1
tunHwFwd **	*boolean*	Hardware forwarding. **Since: 4.21.0
tunIKey **	*integer*	Output key. **Since: 4.20.0
tunMode	*enum: tunnelIntfModeIpsec | tunnelIntfModeUnspecified | tunnelIntfModeGre | tunnelIntfModeIpip*	Tunnel mode.
tunMtu	*integer*	Tunnel transport MTU.
tunOKey **	*integer*	Input key. **Since: 4.20.0
tunPathMtuDiscovery	*boolean*	Tunnel path MTU discovery.
tunSrcAddr	*IP address*	Address for source.
tunTos	*integer*	Tunnel TOS.
tunTtl	*integer*	Tunnel TTL.
tunUnderlayVrf **	*string*	Tunnel underlay VRF. **Since: 4.22.1

2.467.10 VlanInterfaceStatus

bandwidth	*integer*	Bandwidth (bit).
[burnedInAddress]	*MAC address*	Burned-in Address.
description	*string*	Configured interface description.
forwardingModel	*enum: dataLink | quietDataLink | unauthorized | layer1Patch | routed | bridged | recirculation | linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.

hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
[interfaceCounters]	*VlanInterfaceCounters* see 2.467.10.1	Interface Counters and related information.
[interfaceStatistics]	*VlanInterfaceStatistics* see 2.467.10.2	Interface statistics, such as rates.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.

2.467.10.1 VlanInterfaceCounters

[counterRefreshTime]	*floating point number*	UTC of last counter refresh.
[inMulticastOctets]	*integer*	Input multicast octets.
[inMulticastPkts] **	*integer*	Input multicast packets. **Since: 4.20.5
[inOctets]	*integer*	Input octets.
[inPkts] **	*integer*	Input packets. **Since: 4.20.5
[inUcastOctets]	*integer*	Input unicast octets.
[inUcastPkts]	*integer*	Input unicast packets.

[lastClear]	*floating point number*	Last clear of interface counters.
[linkStatusChanges]	*integer*	Number of link status changes since last clear.
[outMulticastOctets]	*integer*	Output multicast octets.
[outMulticastPkts] **	*integer*	Output multicast packets. **Since: 4.20.5
[outOctets]	*integer*	Output octets.
[outPkts] **	*integer*	Output packets. **Since: 4.20.5
[outUcastOctets]	*integer*	Output unicast octets.
[outUcastPkts]	*integer*	Output unicast packets.

2.467.10.2 VlanInterfaceStatistics

inBitsRate	*floating point number*	input bits rate.
outBitsRate	*floating point number*	output bits rate.
updateInterval	*floating point number*	Stats update interval in seconds.

2.467.11 VxlanInterfaceStatus

arpLocalAddress **	*boolean*	Enable rewrite of virtual IP with local IP in VXLAN ARP requests. **Since: 4.20.0
arpReplyRelayMode **	*boolean*	ARP Reply Relay Mode. **Since: 4.18.0
bandwidth	*integer*	Bandwidth (bit).
[bfdPrefixList] **	*string*	Prefix list for filtering BFD sessions. **Since: 4.26.0
[burnedInAddress]	*MAC address*	Burned-in Address.
controllerClientMode	*boolean*	Controller Client Mode.
controllerControlPlane **	*boolean*	Controller Control Plane. **Since: 4.18.1
dataPathLearningMode	*boolean*	Datapath Learning Mode.
decapFilterIntf **	*list of Interface name*	VXLAN decapsulation filter excluded interfaces. **Since: 4.25.0
decapFilterMode **	*enum: filterEnabled* | *filterRelaxedAll* | *filterEnabledIntf* | *filterRelaxedIntf* | *filterDisabled*	VXLAN decapsulation filter mode. **Since: 4.25.0
decapRouteAll **	*boolean*	Routing all VXLAN decasulated packets is enabled. **Since: 4.29.0
[decapVniToVrfMap] **	*table(numeric string) = string*	Decapsulation VNI to VRF mapping. **Since: 4.25.0
description	*string*	Configured interface description.
[detectMult] **	*integer*	Number of consecutive BFD packets missed before failure. **Since: 4.25.2

[floodLearnedAll] **	*boolean*	Flood VTEPs learned via datapath learning. **Since: 4.25.1
forwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
hardware	*enum: loopback* | *vlan* | *ethernet* | *portChannel* | *dps* | *cpu* | *switch* | *application* | *tunnel* | *test* | *internalRecirc* | *multiaccess* | *metamux* | *trafficGen* | *vxlan* | *subinterface* | *fabric* | *metawatch* | *fpgaManagement*	Hardware Description.
interfaceAddress	*list of InterfaceAddressIp4Base* see 2.467.1.1	IPv4 addresses for this interface.
[interfaceAddressIp6]	*InterfaceAddressIp6Base* see 2.467.1.4	IPv6 addresses for this interface.
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
l2Mru **	*integer*	L2 MRU (bytes). **Since: 4.25.0
l3MtuConfigured **	*boolean*	Per-interface MTU is set under interface configuration mode. **Since: 4.23.0
[lastStatusChangeTimestamp]	*floating point number*	Timestamp of last status change.
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.

[maintenanceEnterTime]	*floating point number*	Time at which interface went under maintenance.
[mcastGrpDecap]	*string*	IP multicast group list for decap.
mcastRouting **	*boolean*	Multicast routing is enabled. **Since: 4.22.1
mlagSharedRouterMacAddr **	*MAC address*	MAC address of the VXLAN Shared Router Mac. **Since: 4.21.3
[mlagSrcIpAddr] **	*IPv4 address*	IP address of the MLAG source interface. **Since: 4.21.3
[mlagSrcIpAddrV6] **	*IPv6 address*	IPv6 address of the MLAG source interface. **Since: 4.24.0
[mlagSrcIpIntf] **	*interface name*	Source interface for MLAG VTEP IP. **Since: 4.21.3
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[physicalAddress]	*MAC address*	Physical Layer Address.
portDot1qVniMapping **	*boolean*	VNI mapping to port, dot1q tag pairs. **Since: 4.20.5
[remoteVxlanDomain] **	*boolean*	VXLAN domain is remote. **Since: 4.27.0
[replicationMode]	*string*	Replication/Flood Mode.
[rxInterval] **	*integer*	Rate in microseconds at which BFD control packets are received. **Since: 4.25.2
secUdpPort **	*integer*	VXLANsec UDP port. **Since: 4.29.0
[srcIpAddr]	*string*	IP address of the source interface.
[srcIpAddrV6] **	*IPv6 address*	IPv6 address of the source interface. **Since: 4.24.0
[srcIpIntf]	*string*	Source interface for vxlan.
staticFloodlists **	*boolean*	Static Floodlists Configured. **Since: 4.21.3
[tcpMssV4] **	*integer*	VXLAN TCP MSS value for IPV4 packet. **Since: 4.28.0
[tcpMssV6] **	*integer*	VXLAN TCP MSS value for IPV6 packet. **Since: 4.28.0
[txInterval] **	*integer*	Rate in microseconds at which BFD control packets are sent. **Since: 4.25.2
udpPort **	*integer*	VXLAN UDP port. **Since: 4.29.0
use32BitVni **	*boolean*	VNI length in the VXLAN header. **Since: 4.22.0
[vArpVtepAddr]	*string*	IP address of virtual vtep.
[vArpVtepAddrV6] **	*IPv6 address*	IPv6 address of the virtual VTEPsource interface. **Since: 4.24.1
[vArpVtepSrcIpIntf] **	*interface name*	Source interface for virtual VTEP IP. **Since: 4.24.1

`vccArpProxy` **	*boolean*	ARP proxying when in Vxlan Controller Client Mode. **Since: 4.28.2
`vccImportVlan` **	*string*	Import-enabled VLANs. **Since: 4.23.0
`vccNdProxy` **	*boolean*	Neighbor Discovery proxying when in Vxlan Controller Client Mode. **Since: 4.29.1
`[vlanToLearnRestrict]`	*table(numeric string) = VxlanVtepPrefixes* see 2.467.11.3	Per Vlan Learn list.
`vlanToVniMap`	*table(numeric string) = VniSourcePair* see 2.467.11.1	Vlan to VNI source pair mapping.
`vlanToVtepList`	*table(numeric string) = VxlanRemoteVtepAddrs* see 2.467.11.2	Per Vlan Headend Vtep list.
`vniInDottedNotation`	*boolean*	True if vni is displayed in dotted notation.
`[vrfToVniMap]` **	*table(string) = integer*	VRF to VNI mapping. **Since: 4.18.1
`[vtepAddrMask]` **	*string*	IP address mask assigned to VTEP. **Since: 4.21.0
`vtepSetForSourcePruning` **	*list of IPv4 address*	Source pruning VTEP list. **Since: 4.22.1
`vtepSourcePruningAll` **	*boolean*	Source pruning for all VTEPs. **Since: 4.22.1
`[vtepToSecProfileTotal]` **	*integer*	Total active secure VTEPs. **Since: 4.25.2
`vtepToVtepBridging` **	*boolean*	VTEP to VTEP bridging is enabled. **Since: 4.22.1
`vxlanArpProxy` **	*boolean*	ARP proxying for requests received over VXLAN. **Since: 4.30.0
`vxlanEncapsulation` **	*enum: ipv4 | ipv6 | ipv4AndIpv6*	VXLAN encapsulation type. **Since: 4.24.0
`vxlanNdProxy` **	*boolean*	ND proxying for requests received over VXLAN. **Since: 4.30.0

2.467.11.1 VniSourcePair

`source`	*string*	Mapping source.
`vni`	*integer*	Vni.

2.467.11.2 VxlanRemoteVtepAddrs

`remoteVtepAddr`	*list of IPv4 address*	List of IPv4 vteps.
`remoteVtepAddr6` **	*list of IPv6 address*	List of IPv6 vteps. **Since: 4.23.2

2.467.11.3 VxlanVtepPrefixes

learnFrom	*enum: learnFromAny* | *learnFromList* | *learnFromFloodList* | *learnFromDefault*	Origin of VTEP learn restriction.
prefixList	*list of IP address prefix*	List of vtep (prefixes).

2.468 show interfaces all hardware default

see 2.440

2.469 show interfaces cable detail [...]

interfaces	*table(Interface name) = BaseTCableTestStatus* see 2.469.1	Mapping of base-t interfaces to cable test status.

2.469.1 BaseTCableTestStatus

cableStatus	*CableStatus* see 2.469.1.2	Overall cable health status.
lengthUncertainty	*integer*	Actual length is within this value of the reported length in meters.
pairAStatus	*CablePairStatus* see 2.469.1.1	Status of cable pair A.
pairBStatus	*CablePairStatus* see 2.469.1.1	Status of cable pair B.
pairCStatus	*CablePairStatus* see 2.469.1.1	Status of cable pair C.
pairDStatus	*CablePairStatus* see 2.469.1.1	Status of cable pair D.
testStatus	*TestingStatus* see 2.469.1.3	Status of cable test run.

2.469.1.1 CablePairStatus

lastLengthChange	*floating point number*	Last time cable pair length changed.
lastStateChange	*floating point number*	Last time cable pair state changed.
lengthChanges	*integer*	Number of cable pair length changes.
pairLength	*integer*	Estimated length of the cable pair in meters.
pairState	*enum: inconclusive | intraPairShort | ok | interPairShort | open | busy*	State of cable pair.
stateChanges	*integer*	Number of cable pair state changes.

2.469.1.2 CableStatus

cableState	*enum: faulty | inconclusive | notconnect | ok*	Overall cable state information.
lastStateChange	*floating point number*	Last time cable state changed.
stateChanges	*integer*	Number of cable state changes.

2.469.1.3 TestingStatus

lastStateChange	*floating point number*	Last time test state changed.
numTestRuns	*integer*	Number of cable test runs.
stateChanges	*integer*	Number of test state changes.
testState	*enum: running | completed | failed | untested*	Current testing state.

2.470 show interfaces config-sanity [...]

see 2.467

2.471 show interfaces counters discards [...]

inDiscardsTotal	*integer*	Total input packets discarded.
interfaces	*table(Interface name) = InterfaceCounterDiscards* see 2.471.1	Interface Discard Counters.
[outDiscardsTotal]	*integer*	Total output packets discarded.

2.471.1 InterfaceCounterDiscards

inDiscards	*integer*	Input packets discarded.
[outDiscards]	*integer*	Output packets discarded.

2.472 show interfaces counters drop-precedence [...]

interfaces	*table(Interface name) = IntfDropPrecedence-Counters* see 2.472.1	Mapping between interface and all corresponding drop-precedence counters.

2.472.1 IntfDropPrecedenceCounters

dpCounters	*table(numeric string) = DpCounters* see 2.472.1.1	Mapping between DP ID and its counter data.

2.472.1.1 DpCounters

[inOctets]	*integer*	Ingress octets (bytes).
[inPkts]	*integer*	Ingress packets.
[outOctets]	*integer*	Egress octets (bytes).
[outPkts]	*integer*	Egress packets.

2.473 show interfaces counters egress acl drop [...]

interfaces	*table(Interface name) = EgressAclDropCounters* see 2.473.1	Mapping between egress interface and egress ACL drop counter data. These packets are dropped by the egress packet processor to avoid loops (source port) or double delivery of packets (MLAG).

2.473.1 EgressAclDropCounters

[mlagPackets]	*integer*	MLAG drop packets.
[sourcePruningPackets]	*integer*	Source port pruning packets.

2.474 show interfaces counters egress mtu [...]

destinations	*table(string) = EgressMtuCounters* see 2.474.1	Mapping between destinations and egress MTU counter data. These packets are bigger than the configured Maximum Transmission Unit of the VLAN or routed port. They get recycled back to the CPU to get fragmented and send an "ICMP Fragmentation Needed" message back to the source.

2.474.1 EgressMtuCounters

[droppedPackets]	*integer*	Packets dropped.
[enqueuedPackets]	*integer*	Packets enqueued.
[totalPackets]	*integer*	Total packets.

2.475 show interfaces counters ip [...]

revision 2

interfaces	*table(Interface name) = InterfaceIpv4Ipv6Counters* see 2.475.1	Mapping of an interface to its counters.

2.475.1 InterfaceIpv4Ipv6Counters

[ipv4InOctets]	*integer*	Number of input IPv4 octets.
[ipv4InPkts]	*integer*	Number of input IPv4 packets.
[ipv4OutOctets]	*integer*	Number of output IPv4 octets.
[ipv4OutPkts]	*integer*	Number of output IPv4 packets.
[ipv6InOctets]	*integer*	Number of input IPv6 octets.
[ipv6InPkts]	*integer*	Number of input IPv6 packets.
[ipv6OutOctets]	*integer*	Number of ouput IPv6 octets.
[ipv6OutPkts]	*integer*	Number of ouput IPv6 packets.

2.476 show interfaces counters ip rates [...]

interfaces	*table(Interface name) = InterfaceIpv4Ipv6CounterRates* see 2.476.1	Mapping between an interface and its rate counters.

2.476.1 InterfaceIpv4Ipv6CounterRates

interval	*floating point number*	Interval in seconds.
[ipv4InBitsRate] **	*integer*	Number of input IPv4 bits per second. **Since: **currentRelease**
[ipv4InOctetsRate]	*integer*	Number of input IPv4 octets per second.
[ipv4InPktsRate]	*integer*	Number of input IPv4 packets per second.
[ipv4OutBitsRate] **	*integer*	Number of output IPv4 bits per second. **Since: **currentRelease**
[ipv4OutOctetsRate]	*integer*	Number of output IPv4 octets per second.
[ipv4OutPktsRate]	*integer*	Number of output IPv4 packets per second.
[ipv6InBitsRate] **	*integer*	Number of input IPv6 bits per second. **Since: **currentRelease**
[ipv6InOctetsRate]	*integer*	Number of input IPv6 octets per second.
[ipv6InPktsRate]	*integer*	Number of input IPv6 packets per second.
[ipv6OutBitsRate] **	*integer*	Number of output IPv6 bits per second. **Since: **currentRelease**
[ipv6OutOctetsRate]	*integer*	Number of output IPv6 octets per second.
[ipv6OutPktsRate]	*integer*	Number of output IPv6 packets per second.
[lastUpdateTimestamp]	*floating point number*	Time of last update.

2.477 show interfaces counters queue [...]

egressQueueCounters	*EgressQueueCounters* see 2.477.1	Queue counters on the egress chip.
[ingressQueueCounters] **	*IngressQueueCounters* see 2.477.2	Queue counters on the ingress chip. **Since: 4.23.2
[ingressVoqCounters] **	*IngressVoqCounters* see 2.477.3	VOQ counters on the ingress source. **Since: 4.20.5

2.477.1 EgressQueueCounters

interfaces	*table(Interface name) = EgressQueueDestinationTypeCounters* see 2.477.1.1	Mapping between interface and destination type counter data.

2.477.1.1 EgressQueueDestinationTypeCounters

[bandwidth] **	*integer*	Interface speed in kbps. **Since: 4.22.0
[loadInterval] **	*floating point number*	Interval in seconds over which moving average is calculated. **Since: 4.22.0
mcastQueues	*EgressQueueTrafficClassCounters* see 2.477.1.1.1	Multicast traffic class counters.
ucastQueues	*EgressQueueTrafficClassCounters* see 2.477.1.1.1	Unicast traffic class counters.

2.477.1.1.1 EgressQueueTrafficClassCounters

trafficClasses	*table(string) = EgressQueueDropPrecedenceCounters* see 2.477.1.1.1.1	Mapping between traffic class and per drop precedence counter data. In some cases, some classes might be aggregated (ex: "TC1,4"), or totally aggregated (ex: "all").

2.477.1.1.1.1 EgressQueueDropPrecedenceCounters

dropPrecedences	*table(string) = Counters* see 2.261.1.1.1.2	Mapping between drop precedence and corresponding egress enqueue and drop counter data.
[schedMode] **	*enum: strictPriority | weightedRoundRobin*	Scheduling mode. **Since: 4.22.0
[shapeRate] **	*ShapeRateModel* see 2.477.1.1.1.2	Operational Shape rate. **Since: 4.22.0
[wrrBw] **	*integer*	Operational Weighted-Round-Robin Bandwidth in percent. **Since: 4.22.0

2.477.1.1.1.2 ShapeRateModel

[percent] **	*integer*	Shape rate as percentage of reference (default link bandwidth). **Since: 4.22.1
[percentOf] **	*enum: unknown* | *subinterfaceShapeRate* | *lineRate* | *schedulingGroupShapeRate* | *parentInterfaceShapeRate*	Reference for shape rate percent. **Since: 4.25.2
rate	*integer*	Shape rate.
[shared] **	*boolean*	Shared LAG shaping configured. **Since: 4.21.0
unit	*enum: kbps* | *unspecified* | *pps*	Shape rate unit.

2.477.2 IngressQueueCounters

interfaces	*table(Interface name) = Counters* see 2.261.1.1.1.2	Mapping between interface and counter data across all traffic classes.

2.477.3 IngressVoqCounters

interfaces	*table(Interface name) = IngressVoqTrafficClassCounters* see 2.477.3.1	Mapping between egress interface and traffic class counter data.

2.477.3.1 IngressVoqTrafficClassCounters

trafficClasses	*table(string) = Counters* see 2.261.1.1.1.2	Mapping between traffic class and counter data.

2.478 show interfaces counters queue drop-precedence [...]

interfaces	*table(Interface name) = IntfDropPrecedenceCounters* see 2.478.1	Mapping between interface and all corresponding drop-precedence counters.

2.478.1 IntfDropPrecedenceCounters

drops	*table(numeric string) = Counters* see 2.261.1.1.1.2	Mapping between interface and counter data.

2.479 show interfaces counters queue drops [...]

interfaces	*table(Interface name) = QueueTrafficClassDropsCounters* see 2.479.1	Mapping between egress interface and traffic class drops counter data.

2.479.1 QueueTrafficClassDropsCounters

trafficClasses	*table(string) = IngressEgressQueueDrops-Counters* see 2.479.1.1	Mapping between traffic class, and ingress VOQ and egress queue drops counter data.

2.479.1.1 IngressEgressQueueDropsCounters

egressQueueCounters	*EgressQueueDrop-PrecedenceDropsCounters* see 2.479.1.1.1	Queue drops counters on the egress chip.
[ingressVoqCounters]	*IngressVoqSourceDrops-Counters* see 2.479.1.1.2	VOQ drops counters on the ingress source.

2.479.1.1.1 EgressQueueDropPrecedenceDropsCounters

[countersSum]	*Counters* see 2.261.1.1.1.2	Counters summed over drop precedence and destination type.
[dropPrecedences]	*table(string) = EgressQueueDestination-TypeDropsCounters* see 2.479.1.1.1.1	Mapping between drop precedence and corresponding egress destination type drops counter data.

2.479.1.1.1.1 EgressQueueDestinationTypeDropsCounters

[intfSpeed] **	*integer*	Interface speed. **Since: 4.22.0
[loadInterval] **	*integer*	Load Interval. **Since: 4.22.0
mcastQueues	*Counters* see 2.261.1.1.1.2	Multicast drops counters.
ucastQueues	*Counters* see 2.261.1.1.1.2	Unicast drops counters.

2.479.1.1.2 IngressVoqSourceDropsCounters

countersSum	*Counters* see 2.261.1.1.1.2	Counters summed over all sources.
sources	*table(string) = Counters* see 2.261.1.1.1.2	Mapping between traffic source and drops counter data.

2.480 show interfaces counters queue drops unicast [...]

egressQueueDropCounters	*EgressQueueDropCounters* see 2.480.1	Queue Drop counters on the egress.

2.480.1 EgressQueueDropCounters

interfaces	*table(Interface name) = EgressUcastQueueDrop-Counters* see 2.480.1.1	Mapping between interface and destination type counter data.

2.480.1.1 EgressUcastQueueDropCounters

ucastQueues	*EgressUcastQueueTraffic-ClassCounters* see 2.480.1.1.1	Unicast traffic class counters.

2.480.1.1.1 EgressUcastQueueTrafficClassCounters

trafficClasses	*table(string) = UcastDropCounters* see 2.480.1.1.1.1	Mapping between traffic class and unicast drop counter data.

2.480.1.1.1.1 UcastDropCounters

[droppedBytes]	*integer*	Unicast octets (bytes) dropped.
[droppedPackets]	*integer*	Unicast packets dropped.

2.481 show interfaces counters queue ingress [...]

sources	*table(string) = IngressVoqCounters* see 2.477.3	Mapping between traffic source and VOQ counters.

2.482 show interfaces counters queue rates [...]

egressQueueCounters	*EgressQueueCounters* see 2.477.1	Queue counters on the egress chip.

2.483 show interfaces counters rates [...]

interfaces	*table(Interface name) = InterfaceCountersRateBase* see 2.483.1	Mapping between an interface rate counters and model.

2.483.1 InterfaceCountersRateBase

Could also be an instance of:
InterfaceCountersRate 2.483.4
VlanInterfaceCountersRate 2.483.5
VxlanInterfaceCountersRate 2.483.6
InterfaceIpv4Ipv6CounterRates 2.476.1
FabricIntfCountersRateModel 2.483.2
FabricPairCountersRateModel 2.483.3
FabricPeerIntfCountersRateModel 2.483.3.1

2.483.2 FabricIntfCountersRateModel

inBpsRate	*floating point number*	Input bps rate.
inPktsRate	*floating point number*	Input packets rate in percentage of port capacity.
inPpsRate	*floating point number*	Input pps rate.
outBpsRate	*floating point number*	Output bps rate.
outPktsRate	*floating point number*	Output packets rate in percentage of port capacity.
outPpsRate	*floating point number*	Output Kpps rate.

2.483.3 FabricPairCountersRateModel

inBpsRate	*floating point number*	Input bps rate.
inPktsRate	*floating point number*	Input packets rate in percentage of port capacity.
inPpsRate	*floating point number*	Input pps rate.
interval	*integer*	Interval in seconds.
outBpsRate	*floating point number*	Output bps rate.
outPktsRate	*floating point number*	Output packets rate in percentage of port capacity.
outPpsRate	*floating point number*	Output Kpps rate.
[peer]	*FabricPeerIntfCounters-RateModel* see 2.483.3.1	Peer interface rates.

2.483.3.1 FabricPeerIntfCountersRateModel

inBpsRate	*floating point number*	Input bps rate.
inPktsRate	*floating point number*	Input packets rate in percentage of port capacity.
inPpsRate	*floating point number*	Input pps rate.
outBpsRate	*floating point number*	Output bps rate.

outPktsRate	*floating point number*	Output packets rate in percentage of port capacity.
outPpsRate	*floating point number*	Output Kpps rate.

2.483.4 InterfaceCountersRate

description	*string*	Port description.
inBpsRate	*floating point number*	Input bps rate.
[inPktsRate]	*floating point number*	Input packets rate percentage.
inPpsRate	*floating point number*	Input pps rate.
interval	*integer*	Interval in seconds.
[lastUpdateTimestamp] **	*floating point number*	Time of last update. **Since: 4.21.3
outBpsRate	*floating point number*	Output bps rate.
[outPktsRate]	*floating point number*	Output packets rate percentage.
outPpsRate	*floating point number*	Output Kpps rate.

2.483.5 VlanInterfaceCountersRate

description	*string*	Port description.
inBpsRate	*floating point number*	Input bps rate.
interval	*integer*	Interval in seconds.
outBpsRate	*floating point number*	Output bps rate.

2.483.6 VxlanInterfaceCountersRate

vnis	*table(string) = VxlanInterfaceVniCountersRate* see 2.483.6.1	Mapping between VNI and its counters rate.

2.483.6.1 VxlanInterfaceVniCountersRate

description	*string*	Port description.
inBpsRate	*floating point number*	Input bps rate.
[inPktsRate]	*floating point number*	Input packets rate percentage.
inPpsRate	*floating point number*	Input pps rate.
interval	*integer*	Interval in seconds.
[lastUpdateTimestamp]	*floating point number*	Time of last update.
outBpsRate	*floating point number*	Output bps rate.
[outPktsRate]	*floating point number*	Output packets rate percentage.
outPpsRate	*floating point number*	Output pps rate.

2.484 show interfaces counters traffic-class [...]

interfaces	*table(Interface name) = InterfaceTrafficClassesCounters* see 2.484.1	A mapping between an interface and its traffic class counters.

2.484.1 InterfaceTrafficClassesCounters

trafficClasses	*table(numeric string) = InterfaceTrafficClassCounters* see 2.484.1.1	A mapping between a traffic class (in range 0 to 7) and its counters.

2.484.1.1 InterfaceTrafficClassCounters

[inBcastPkts]	*integer*	Input broadcast packets.
[inDroppedMcastOctets] **	*integer*	Dropped multicast octets. **Since: 4.30.0
[inDroppedMcastPkts] **	*integer*	Dropped multicast packets. **Since: 4.30.0
[inDroppedUcastOctets] **	*integer*	Dropped unicast octets. **Since: 4.30.0
[inDroppedUcastPkts] **	*integer*	Dropped unicast packets. **Since: 4.30.0
[inMcastOctets] **	*integer*	Input multicast octets. **Since: 4.30.0
[inMcastPkts]	*integer*	Input multicast packets.
inOctets	*integer*	Input octets.
inPkts	*integer*	Input packets.
[inUcastOctets] **	*integer*	Input unicast octets. **Since: 4.30.0
[inUcastPkts]	*integer*	Input unicast packets.

2.485 show interfaces counters vni [...]

Could also be an instance of:
VxlanInterfacesVniCounters 2.485

interfaces	*table(Interface name) = VirtualInterfaceCounters-Base* see 2.485.1	Interface Counters.

2.485.1 VirtualInterfaceCountersBase

Could also be an instance of:
VxlanInterfaceCounters 2.485.2

2.485.2 VxlanInterfaceCounters

vnis	*table(string) = VxlanInterfaceVniCounters* see 2.485.2.1	Mapping between VNI and its counters.

2.485.2.1 VxlanInterfaceVniCounters

inOctets	*integer*	Input octets.
inPackets	*integer*	Input packets.
[lastUpdateTimestamp]	*floating point number*	Time of last update.

outOctets	*integer*	Output octets.
outPackets	*integer*	Output packets.

2.486 show interfaces counters vni rates [...]

see 2.483

2.487 show interfaces counters | (counters (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfaceCountersBase* see 2.487.1	Interface Counters.

2.487.1 InterfaceCountersBase

Could also be an instance of:
InterfaceCounters 2.487.2
VirtualInterfaceCountersBase 2.485.1
EthInterfaceCounters 2.467.3.1
VlanInterfaceCounters 2.467.10.1
VxlanInterfaceCounters 2.485.2
InterfaceFastCounters 2.487.3

2.487.2 InterfaceCounters

inBroadcastPkts	*integer*	Input broadcast packets.
[inDiscards]	*integer*	Input packets with errors.
inMulticastPkts	*integer*	Input multicast packets.
inOctets	*integer*	Input octets.
[inTotalPkts] **	*integer*	Input packets. **Since: 4.20.0
inUcastPkts	*integer*	Input unicast packets.
[lastUpdateTimestamp] **	*floating point number*	Time of last update. **Since: 4.21.3
outBroadcastPkts	*integer*	Output broadcast packets.
[outDiscards]	*integer*	Output packets with errors.
outMulticastPkts	*integer*	Output multicast packets.
outOctets	*integer*	Output octets.
[outTotalPkts] **	*integer*	Output packets. **Since: 4.28.2
outUcastPkts	*integer*	Output unicast packets.

2.487.3 InterfaceFastCounters

[inBroadcastPkts]	*integer*	Input broadcast packets.
[inMulticastPkts]	*integer*	Input multicast packets.
[inOctets]	*integer*	Input octets.
[inUcastPkts]	*integer*	Input unicast packets.
lastUpdateTimestamp	*floating point number*	Time of last update.
[outBroadcastPkts]	*integer*	Output broadcast packets.
[outMulticastPkts]	*integer*	Output multicast packets.
[outOctets]	*integer*	Output octets.

[outUcastPkts]	*integer*	Output unicast packets.

2.488 show interfaces debounce | (debounce (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfaceDebounceTime* see 2.488.1	Mapping between an interface and debounce times.

2.488.1 InterfaceDebounceTime

linkDownTime	*integer*	Interface linkdown debounce time in milliseconds, 0 means disabled.
linkUpTime	*integer*	Interface linkup debounce time in milliseconds, 0 means disabled.

2.489 show interfaces description [...]

interfaceDescriptions	*table(Interface name) = InterfaceDescription* see 2.489.1	Mapping between an interface and its description.

2.489.1 InterfaceDescription

description	*string*	Configured interface description.
interfaceStatus	*enum: down | adminDown | notApplicable | uninitDown | up*	Connection status of the interface.
lineProtocolStatus	*enum: notPresent | lowerLayerDown | unknown | testing | notApplicable | down | dormant | up*	Line protocol status.

2.490 show interfaces encapsulation dot1q vlan [...]

intfEncapsulations	*table(string) = InterfaceEncapsulations* see 2.490.1	The currently configured encapsulation vlan, keyed by interface name.

2.490.1 InterfaceEncapsulations

egressActive	*boolean*	Whether the encapsulation is active on egress.
ingressActive	*boolean*	Whether the encapsulation is active on ingress.
vlanId	*string*	The dot1q encapsulation VLAN Id.

2.491 show interfaces encapsulation vlan [...]

intfEncapsulations	*table(string) = FlexibleEncapsulations* see 2.491.1	Interface VLAN encapsulation status.

2.491.1 FlexibleEncapsulations

entries	*list of FlexEncapEntry* see 2.491.1.1	VLAN encapsulation entries.

2.491.1.1 FlexEncapEntry

client	*FlexEncapSpec* see 2.491.1.1.1	Client side encapsulation.
clientId **	*string*	Client identifier for encapsulation entries. **Since: 4.30.0
[network]	*FlexEncapSpec* see 2.491.1.1.1	Network side encapsulation.
status	*enum: inactive | conflict | overlap | active | asymmetric*	VLAN encapsulation status.

2.491.1.1.1 FlexEncapSpec

specType	*enum: clientInner | tagged | untagged | client | unmatched*	Type of encapsulation.
[tagged]	*FlexEncapTaggedSpec* see 2.491.1.1.1.1	Tagged encapsulation.

2.491.1.1.1.1 FlexEncapTaggedSpec

[inner]	*FlexEncapValue* see 2.491.1.1.1.2	Inner VLAN tags.
outer	*FlexEncapValue* see 2.491.1.1.1.2	Outer VLAN tags.

2.491.1.1.1.2 FlexEncapValue

[tags] **	*list of FlexEncapTaggedValue* see 2.491.1.1.1.3	VLAN tags. **Since: 4.24.2
valueType	*enum: dot1q | dot1ad | client*	Type of encapsulation.

2.491.1.1.1.3 FlexEncapTaggedValue

[endTag] **	*integer*	Ending VLAN tag. **Since: 4.29.1
[tag]	*integer*	Starting VLAN tag.

2.492 show interfaces interactions | (interactions (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaces	*table(Interface name) = InterfacesInteractionsDataBySpeed* see 2.492.1	A mapping of interface names to interactions with other interfaces at a desired speed configuration.

2.492.1 InterfacesInteractionsDataBySpeed

[inactiveReason] **	*string*	Reason why the interface is inactive due to having no available configurations. **Since: **currentRelease**
[interfaceHardwareResourcePool] **	*InterfaceHardwareResourcePool* see 2.492.1.1	Interface hardware resource pool that this interface belongs to. **Since: 4.23.1
speeds	*table(string) = InterfacesInteractionsData* see 2.492.1.2	A mapping of desired speed configurations to interactions with other interfaces. Speed configurations are represented in the notation <speed> or <speed> -<lane> . <speed> describes the interface speed and, <lane> if specified describes the number of electrical lanes used on the host-side of the transceiver.

2.492.1.1 InterfaceHardwareResourcePool

capacity	*integer*	Maximum number of interface hardware resources available in the pool.
memberInterfaces	*list of Interface name*	Interfaces belonging to this resource pool.
poolId	*integer*	Name of the resource pool.

2.492.1.2 InterfacesInteractionsData

[autonegRequirement] **	*AutonegFecRequirement* see 2.492.1.2.1	Error-correct encoding restriction when auto-negotation is configured. **Since: 4.26.2
[compatibleParentInterfaceConfigurations] **	*CompatibleParentInterfaceConfigurations* see 2.492.1.2.2	List of interfaces that can affect desired configuration and which of their configurations are compatible with desired configuration. **Since: 4.23.1
[hardwarePortGroupRequirement] **	*HardwarePortGroupRequirement* see 2.492.1.2.3	Hardware port-group configuration required to achieve stated speed configuration. **Since: 4.27.0
[inactiveInterfaces]	*InactiveInterfacesInteraction* see 2.492.1.2.4	List of interfaces that become inactive to facilitate desired speed configuration.

[primaryInterface-ConfigurationRequirement]	*PrimaryInterface-ConfigurationRequirement* see 2.492.1.2.5	Speed configuration required on primary interface to facilitate desired speed configuration.
[speedGroupRequirement] **	*SpeedGroupRequirement* see 2.492.1.2.6	Speed-group configuration required to achieve stated speed configuration. **Since: 4.22.1
[speedLimitedInterfaces]	*SpeedLimitedInterfaces-Interaction* see 2.492.1.2.7	List of interfaces that become speed-limited and what speeds they are limited to to facilitate desired speed configuration.

2.492.1.2.1 AutonegFecRequirement

primaryIntf	*interface name*	Name of the primary interface.

2.492.1.2.2 CompatibleParentInterfaceConfigurations

interfaces	*table(Interface name) = SpeedConfigurationList* see 2.492.1.2.2.1	Mapping of interface name to a slash-separated list of its speed configurations that do not conflict with the desired configuration on the desired interface.

2.492.1.2.2.1 SpeedConfigurationList

configurations	*list of SpeedConfiguration* see 2.492.1.2.2.2	List of speed configurations.

2.492.1.2.2.2 SpeedConfiguration

[lanes]	*integer*	Number of electrical lanes used on the host-side of the interface.
speed	*enum: 100G | 25G | 100M | 5G | 200G | 50G | 10M | 10G | 2.5G | 400G | 1G | 800G | 40G*	Configuration speed.

2.492.1.2.3 HardwarePortGroupRequirement

portGroup	*string*	Port-group the referred interface is in.
portMode	*string*	Port-group mode needed for the referred interface speed to take effect.

2.492.1.2.4 InactiveInterfacesInteraction

interfaces	*list of Interface name*	Interfaces that become inactive.

2.492.1.2.5 PrimaryInterfaceConfigurationRequirement

interface	*interface name*	Name of primary interface.

requiredConfiguration	*enum: 25G | 100G-1 | 50G-2 | 100M | 800G-8 | 200G-4 | 200G-8 | 400G-4 | 200G-2 | 100G-4 | 10G | 10M | 400G-8 | 100G-2 | 1G | 50G-1 | 40G*	Speed configuration required on primary interface. Speed configuration is given in the notation <speed> or <speed> -<lane> . <speed> describes the interface speed, and <lane> if specified describes the number of electrical lanes used on the host-side of the transceiver.

2.492.1.2.6 SpeedGroupRequirement

serdes	*string*	Speed-group SerDes configuration needed for the referred interface speed to take effect.
speedGroup	*string*	Speed-group the referred interface is in.

2.492.1.2.7 SpeedLimitedInterfacesInteraction

interfaces	*table(Interface name) = string*	Mapping of interface name to what speed configurations it becomes limited to. Speed configurations are represented in the notation <speed> or <speed> -<lane> . <speed> describes the interface speed, and <lane> if specified describes the number of electrical lanes used on the host-side of the transceiver.

2.493 show interfaces mac [...]

interfaces	*table(Interface name) = InterfaceMacData* see 2.493.1	A mapping of interfaces to their MAC information.

2.493.1 InterfaceMacData

[additionalStatus]	*table(string) = string*	A mapping of platform specific fields to their values.
adminEnabled	*boolean*	Administratively enabled.
intfState	*enum: linkUp | linkUnknown | linkDown | inactive*	Interface State.
intfStateChangeCount	*integer*	Total number of interface state changes.
intfStateLastChangeTime	*floating point number*	Time of last interface state change.
macRxLastLocalFaultChange-Time	*floating point number*	Time of last local MAC Rx fault.
macRxLastRemoteFaultChange-Time	*floating point number*	Time of last remote MAC Rx fault.
macRxLocalFault	*boolean*	A current local Rx fault has been detected.
macRxLocalFaultChangeCount	*integer*	Total number of local MAC Rx faults.
macRxRemoteFault	*boolean*	A current remote Rx fault has been detected.
macRxRemoteFaultChangeCount	*integer*	Total number of remote MAC Rx faults.

[phyState]	*enum: errDisabled | unknown | linkUp | linkDown | detectingXcvr | init | autonegotiating*	Phy State.
[phyStateChangeCount]	*integer*	Total number of PHY state changes.
[phyStateLastChangeTime]	*floating point number*	Time of last PHY state change.

2.494 show interfaces queue length [...]

cpuPorts	*table(Interface name) = QueueTypeQueueInfo* see 2.494.1	Maps CPU port name to queue information.
fabricPorts	*table(Interface name) = QueueTypeQueueInfo* see 2.494.1	Maps fabric port name to queue information.
frontPanelPorts	*table(Interface name) = QueueTypeQueueInfo* see 2.494.1	Maps front panel port name to queue information.
lagSubIntfs **	*table(Interface name) = QueueTypeQueueInfo* see 2.494.1	Maps LAG sub interface name to queue information. **Since: 4.22.0
recyclePorts	*table(Interface name) = QueueTypeQueueInfo* see 2.494.1	Maps recycle port name to queue information.

2.494.1 QueueTypeQueueInfo

Could also be an instance of:
SandQueueTypeQueueInfo 2.494.2

2.494.2 SandQueueTypeQueueInfo

voqs	*table(string) = SourceQueueInfo* see 2.494.2.1	Maps traffic class to queue information for virtual output queue type.

2.494.2.1 SourceQueueInfo

sources	*table(string) = QueueInfo* see 2.494.2.1.1	Maps traffic source name to queue information.

2.494.2.1.1 QueueInfo

queueLength	*integer*	Current queue length in bytes.

2.495 show interfaces single INTF

see 2.467

2.496 show interfaces status | (status (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

interfaceStatuses	*table(string) = InterfaceStatusEntry* see 2.496.1	Collection of interfaces statuses, keyed by interface name.

2.496.1 InterfaceStatusEntry

[autoNegotiateActive]	*boolean*	Is autonegotiate active.
autoNegotigateActive	*boolean*	Is autonegotiate active. Misspelled autoNegotiateActive (legacy).
bandwidth	*integer*	Bandwidth (bit).
description	*string*	Configured interface description.
duplex	*enum: duplexHalf | duplexUnknown | duplexFull*	Duplex status of the interface.
[encapsulation] **	*Encapsulation* see 2.496.1.1	Encapsulation information. **Since: 4.21.0
interfaceType	*string*	Interface Type.
[lineProtocolStatus]	*enum: notPresent | lowerLayerDown | unknown | testing | notApplicable | down | dormant | up*	Line protocol status.
linkStatus	*enum: signal | inactive | connected | unknown | maint-down | errdisabled | disabled | admin | notapplicable | notconnect | noSignal | initializing | uninitialized*	Connection status of the interface.
[trafficAboveThreshold]	*boolean*	Indicates that the traffic on the Interface is above traffic threshold.
[underMaintenance]	*boolean*	Indicates an Interface is under maintenance or not.
vlanInformation	*VlanInformation* see 2.496.1.2	Model to describe the VLAN setting.

2.496.1.1 Encapsulation

[dot1qVlanTags]	*list of integer*	Encapsulation dot1q vlan tags.
encapsulationType	*dot1q*	Encapsulation type.

2.496.1.2 VlanInformation

interfaceForwardingModel	*enum: dataLink* | *quietDataLink* | *unauthorized* | *layer1Patch* | *routed* | *bridged* | *recirculation* | *linkQualification*	LinkQualification – the interface forwards packets for link qualification purposes. Protocols do not run on this interface, Layer1Patch – the interface runs in Layer 1 mode to patch receive and/or transmit paths of various ports in the system. Protocols do not run on this interface, Recirculation – the interface recirculates packets. Protocols do not run on this port, QuietDataLink – the interface forwards packets but does not participate in bridging or routing and protocols do not run on this interface, Unauthorized – the interface does not participate in bridging or routing and only EAPOL runs on this interface, DataLink – the interface forwards packets given to it but is not participating in bridging or routing operations, Routed – the interface can route packets, Bridged – the interface can bridge packets but not route them.
[interfaceMode]	*enum: inactive* | *routed* | *tap* | *trunk* | *bridged* | *tapTool* | *tool* | *encap*	If not in DataLink mode, what mode.
[vlanExplanation]	*string*	Shows whether the interface is part of a monitor destination or a Port-channel.
[vlanId]	*integer*	Vlan ID if interface is bridged, None otherwise.

2.497 show interfaces switchport configuration source [...]

interfaces	*table(Interface name)* = *ConfigSource* see 2.497.1	Switchport config source indexed by interface name.

2.497.1 ConfigSource

config	*list of string*	List of config commands published by this source.
source	*string*	Name of the config source.

2.498 show interfaces switchport vlan mapping [...]

intfVlanMappings	*table(string) = InterfaceVlanMappings* see 2.498.1	Collection of VLAN mappings, keyed by interface.

2.498.1 InterfaceVlanMappings

[egressDoubleTagged-Mappings]	*table(numeric string) = InnerTagMappings* see 2.498.1.1	Double tagged VLAN mappings configured for egress packets on an interface.
egressVlanMappingRequired **	*boolean*	Whether VLAN mapping is required for egress packets. **Since: 4.29.1
egressVlanMappings	*table(numeric string) = MappedVlan* see 2.498.1.1.1	VLAN mappings configured for egress packets on an interface.
[ingressDoubleTagged-Mappings]	*table(numeric string) = InnerTagMappings* see 2.498.1.1	Double tagged VLAN mappings configured for ingress packets on an interface.
ingressVlanMappingRequired **	*boolean*	Whether VLAN mapping is required for ingress packets. **Since: 4.20.5
ingressVlanMappings	*table(numeric string) = MappedVlan* see 2.498.1.1.1	VLAN mappings configured for ingress packets on an interface.

2.498.1.1 InnerTagMappings

mappings	*table(numeric string) = MappedVlan* see 2.498.1.1.1	Inner tag VLAN mappings.

2.498.1.1.1 MappedVlan

active	*boolean*	Whether this mapping is currently active.
[allInnerVid] **	*boolean*	Match on all inner VlanId for dot1q tunnel packet. **Since: 4.27.1
configured	*boolean*	Whether this mapping is currently configured.
[deQinQXlate] **	*boolean*	Untunnel and translate double tagged packets. **Since: 4.27.1
[dot1qTunnel]	*boolean*	Whether the tag is dot1q tunneled.
[innerVlanId]	*integer*	Inner VLAN ID that packets are mapped to.
vlanId	*integer*	VLAN ID that packets are mapped to (the outer VLAN in a double mapping).

2.499 show interfaces switchport | (switchport (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

switchports	*table(Interface name) = SwitchPort* see 2.499.1	A collection of switchports indexed by interface name.

2.499.1 SwitchPort

enabled	*boolean*	Whether switchport is enabled.
[switchportInfo]	*SwitchPortInfo* see 2.499.1.1	Switchport information.

2.499.1.1 SwitchPortInfo

accessVlanId	*integer*	Access VLAN ID.
accessVlanName	*string*	Access VLAN name.
dot1qVlanTagDisallowed **	*boolean*	Drop tagged frames is active on the port. **Since: 4.21.3
dot1qVlanTagRequired **	*boolean*	Allow only vlan tagged frames. **Since: 4.20.5
dot1qVlanTagRequiredStatus **	*boolean*	Allow only vlan tagged frames is active on the port. **Since: 4.20.5
dynamicAllowedVlans	*table(string) = string*	A mapping between a process name and dynamic VLANs out of the allowed trunk VLANs on the switchport. Represented in canonical form.
dynamicBlockedVlans **	*table(string) = string*	A mapping between a process name and VLANs blocked by this process on the switchport. Represented in canonical form. **Since: 4.26.0
dynamicTrunkGroups	*list of string*	List of dynamic trunk group names.
macLearning	*boolean*	Whether MAC address learning is enabled.
mbvaEnabled **	*boolean*	MAC-based VLAN assignment is enabled. **Since: 4.27.1
mode	*enum: tap | trunk | tapTool | tool | dot1qTunnel | access*	Administrative or operational mode of the switchport.
phoneTrunk **	*boolean*	Phone Trunk mode is enabled. **Since: 4.22.1
phoneTrunkUntagged **	*boolean*	Phone trunk untagged is enabled. **Since: 4.22.1
phoneVlan **	*integer*	Phone VLAN ID. **Since: 4.22.1
[privateVlanMapping]	*string*	Administrative private VLANs mapped to the switchport. Represented in canonical form.

sourceportFilterMode **	*enum: disabled | enabled | multicastDisabled*	Whether source interface filtering is disabled or not. **Since: 4.18.1
[staticAllowedVlans]	*string*	Static VLANs out of the allowed trunk VLANs on the switchport. Represented in canonical form.
staticTrunkGroups	*list of string*	List of static trunk group names.
[tpid]	*string*	Dot1q ethertype/Tag protocol ID in hex.
[tpidStatus]	*boolean*	Whether tpid is active or not.
trunkAllowedVlans	*string*	Allowed trunk VLANs on the switchport. Represented in canonical form. For eg: (2-40,70,100-200), ALL(1-4094), NONE etc.
trunkingNativeVlanId	*integer*	Trunking native mode VLAN ID.
[trunkingNativeVlanName]	*string*	Trunking native mode VLAN name.
vlanForwardingMode **	*enum: allConfiguredVlans | allowedVlansOnly*	Whether the VLAN forwarding mode is set to allowed VLANs on the port or all configured VLANs. **Since: 4.21.3

2.500 show interfaces transceiver [...]

interfaces	*table(Interface name) = InterfaceTransceiverBase* see 2.500.1	Mapping between Interface name and the transceiver info.

2.500.1 InterfaceTransceiverBase

Could also be an instance of:
InterfaceTransceiver 2.500.2

2.500.2 InterfaceTransceiver

[details]	*InterfaceTransceiverDetails* see 2.500.2.1	Warning and alarm thresholds.
mediaType	*string*	Media type.
[narrowBand] **	*boolean*	Narrowband Transceiver. **Since: 4.15.4
[rxPower]	*floating point number*	RX power in dBm.
[temperature]	*floating point number*	Temperature in Celsius.
[totalRxPower] **	*floating point number*	Total RX power in dBm. **Since: 4.15.4
[txBias]	*floating point number*	Current in mA.
[txPower]	*floating point number*	TX power in dBm.
updateTime	*floating point number*	Last update time in UTC.
vendorSn	*string*	Transceiver serial number.
[voltage]	*floating point number*	Voltage in Volts.

2.500.2.1 InterfaceTransceiverDetails

[rxPower]	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	RX power alarm and warning thresholds in dBm.
[temperature]	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	Temperature alarm and warning thresholds in Celsius.
[totalRxPower] **	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	Total RX power alarm and warning thresholds in dBm. **Since: 4.15.4
[txBias]	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	Bias current alarm and warning thresholds in mA.
[txPower]	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	TX power alarm and warning thresholds in dBm.
[voltage]	*InterfaceTransceiverDetailThresholds* see 2.500.2.1.1	Voltage alarm and warning thresholds in Volts.

2.500.2.1.1 InterfaceTransceiverDetailThresholds

[highAlarm]	*floating point number*	High alarm threshold for parameter.
[highWarn]	*floating point number*	High warning threshold for parameter.
[lowAlarm]	*floating point number*	Low alarm threshold for parameter.
[lowWarn]	*floating point number*	Low warning threshold for parameter.

2.501 show interfaces transceiver diag test pattern [...]

lineInterfaces	*table(Interface name) = InterfacesTransceiverTestPatternConfiguration* see 2.501.1	Mapping from interface name to line/media side transceiver Test Pattern config information.
systemInterfaces	*table(Interface name) = InterfacesTransceiverTestPatternConfiguration* see 2.501.1	Mapping from interface name to system/host side transceiver Test Pattern config information.

2.501.1 InterfacesTransceiverTestPatternConfiguration

receiver	*InterfacesTransceiverTestPatternSideConfigs* see 2.501.1.1	Configuration for receiver / checker.
transmitter	*InterfacesTransceiverTestPatternSideConfigs* see 2.501.1.1	Configuration for transmitter / generator.

2.501.1.1 InterfacesTransceiverTestPatternSideConfigs

[configuredOptions]	*InterfacesTransceiverTestPatternSideConfig* see 2.501.1.1.1	Test Pattern options selected by the user.
[operationalOptions]	*InterfacesTransceiverTestPatternSideConfig* see 2.501.1.1.1	Test Pattern options currently operational in the module.

2.501.1.1.1 InterfacesTransceiverTestPatternSideConfig

fec	*boolean*	Forward error correction enabled in this config.
pattern	*enum: prbs31q | prbs9q | prbs15q | prbs9 | prbs31 | prbs13q | prbs23 | prbs23q | prbs7 | prbs7q | ssprq | prbs13 | prbs15*	The pattern present in this config.

2.502 show interfaces transceiver diag test pattern capabilities [...]

interfaces	*table(Interface name) = InterfacesTransceiverTestPatternCapabilities* see 2.502.1	Mapping between interface name and the transceiver info.

2.502.1 InterfacesTransceiverTestPatternCapabilities

[lineReceiver]	*InterfacesTransceiverTestPatternSideCapabilities* see 2.502.1.1	Capabilities for line receiver / media checker.
[lineTransmitter]	*InterfacesTransceiverTestPatternSideCapabilities* see 2.502.1.1	Capabilities for line transmitter / media generator.
[systemReceiver]	*InterfacesTransceiverTestPatternSideCapabilities* see 2.502.1.1	Capabilities for system receiver / host checker.
[systemTransmitter]	*InterfacesTransceiverTestPatternSideCapabilities* see 2.502.1.1	Capabilities for system transmitter / host generator.

2.502.1.1 InterfacesTransceiverTestPatternSideCapabilities

fecSupported	*boolean*	Does this module support forward error correction during test pattern testing.
[supportedPatterns]	*list of string*	The possible patterns that the module's diagnostics can generate and do error counting for.

2.503 show interfaces transceiver diag test pattern counters [...]

lineInterfaces	*table(Interface name) = InterfacesTransceiverTestPatternCounters* see 2.503.1	List of line/media side transceiver test pattern result counters indexed by lane number.
systemInterfaces	*table(Interface name) = InterfacesTransceiverTestPatternCounters* see 2.503.1	List of system/host side transceiver test pattern result counters indexed by lane number.

2.503.1 InterfacesTransceiverTestPatternCounters

[lanes]	*list of InterfacesTransceiverTestPatternLaneCounters* see 2.503.1.1	Mapping from lane number to test pattern result lane counters.
[lastClear]	*floating point number*	The time when the pattern was last cleared.
[operationalPattern]	*enum: prbs31q | prbs9q | prbs15q | prbs9 | prbs31 | prbs13q | prbs23 | prbs23q | prbs7 | prbs7q | ssprq | prbs13 | prbs15*	The active test pattern.

2.503.1.1 InterfacesTransceiverTestPatternLaneCounters

bitErrors	*integer*	The number of bit errors since test pattern was enabled.
burstCount	*integer*	The number of polling periods where counters changed since test pattern was enabled.

[detailed]	*InterfacesTransceiverTestPatternLaneCountersDetail* see 2.503.1.1.1	Detailed test pattern lane counters.
largestBurst	*integer*	The largest amount of bit errors in a polling period since test pattern was enabled.
[lastErrorTime]	*floating point number*	The last time an error was reported since test pattern was enabled.
linkState	*enum: locked | notLocked | notApplicable*	The lock state of the lane link under test.

2.503.1.1.1 InterfacesTransceiverTestPatternLaneCountersDetail

ber	*floating point number*	The bit error rate of the link.
[lastLinkStateChange]	*floating point number*	The time when the link state last changed.
linkStateChanges	*integer*	The number of link state transitions since test pattern was enabled.
totalBits	*integer*	The total number of bits sent through the link.

2.504 show interfaces transceiver dom [...]

interfaces	*table(Interface name) = InterfaceTransceiverDomBase* see 2.504.1	Mapping between interface name and the transceiver information.

2.504.1 InterfaceTransceiverDomBase

Could also be an instance of:
InterfaceTransceiverDom 2.504.2

[displayName] **	*string*	Interfaces corresponding to the channel of the transceiver. **Since: 4.22.0

2.504.2 InterfaceTransceiverDom

[displayName] **	*string*	Interfaces corresponding to the channel of the transceiver. **Since: 4.22.0
mediaType	*string*	Media type.
parameters	*table(string) = InterfaceTransceiverDomParameter* see 2.504.2.1	A mapping of parameter name to per channel value and thresholds.
updateTime	*floating point number*	Last update time in UTC.
vendorSn	*string*	Transceiver serial number.

2.504.2.1 InterfaceTransceiverDomParameter

channels	*table(string) = floating point number or NULL*	Per channel parameter value. '-' for the channel indicates a channel independent parameter.
[threshold]	*InterfaceTransceiverDetail-Thresholds* see 2.500.2.1.1	Threshold for monitored parameter.
unit	*string*	Parameter unit.

2.505 show interfaces transceiver eeprom [...]

interfaces	*table(Interface name) = InterfacesTransceiverEepromIntf* see 2.505.1	Mapping between interface name and the EEPROM.

2.505.1 InterfacesTransceiverEepromIntf

fields	*table(string) = InterfacesTransceiverEepromField* see 2.505.1.1	Mapping between a field name and its data.

2.505.1.1 InterfacesTransceiverEepromField

[note]	*string*	A note about this field's interpretation.
pages	*table(numeric string) = InterfacesTransceiverEepromBytes* see 2.505.1.1.1	Mapping between a page and the bytes on the page the field is defined over.
[subfields]	*table(string) = InterfacesTransceiverEepromFieldBase* see 2.505.1.1.2	Mapping between a subfield name and its data.
[value]	*InterfacesTransceiverEepromValue* see 2.505.1.1.3	The value of this field.

2.505.1.1.1 InterfacesTransceiverEepromBytes

byteOffsets	*list of integer*	The byte offsets for the field.

2.505.1.1.2 InterfacesTransceiverEepromFieldBase

Could also be an instance of:
InterfacesTransceiverEepromField 2.505.1.1

2.505.1.1.3 InterfacesTransceiverEepromValue

Could also be an instance of:
InterfacesTransceiverEepromBoolValue 2.505.1.1.4
InterfacesTransceiverEepromFloatValue 2.505.1.1.5
InterfacesTransceiverEepromIntValue 2.505.1.1.6
InterfacesTransceiverEepromStrValue 2.505.1.1.7

raw	*integer*	The raw value stored in EEPROM.

2.505.1.1.4 InterfacesTransceiverEepromBoolValue

decoded	*boolean*	The decoded value from the EEPROM.

raw	*integer*	The raw value stored in EEPROM.

2.505.1.1.5 InterfacesTransceiverEepromFloatValue

decoded	*floating point number*	The decoded value from the EEPROM.
[prefix]	*enum: +/- | +* |	Prefix such as "+/-" for the decoded value.
raw	*integer*	The raw value stored in EEPROM.
[units]	*enum: ms | m | dB | C | s | MHz | MBd | Mbps | km | nm | W*	The units for the decoded value.

2.505.1.1.6 InterfacesTransceiverEepromIntValue

decoded	*integer*	The decoded value from the EEPROM.
[prefix]	*enum: +/- | +* |	Prefix such as "+/-" for the decoded value.
raw	*integer*	The raw value stored in EEPROM.
[units]	*enum: ms | m | dB | C | s | MHz | MBd | Mbps | km | nm | W*	The units for the decoded value.

2.505.1.1.7 InterfacesTransceiverEepromStrValue

decoded	*string*	The decoded value from the EEPROM.
[possibleValues]	*list of string*	The possible values that decoded can take on.
raw	*integer*	The raw value stored in EEPROM.

2.506 show interfaces transceiver hardware [...]

revision 2

interfaces	*table(Interface name) = InterfacesTransceiverHardware* see 2.506.1	Mapping between interface name and the transceiver info.

2.506.1 InterfacesTransceiverHardware

[cableType]	*enum: CA-S | CA-N | CA-L*	Type of copper cable.
[detectedMediaType]	*string*	The media type set in device's IDPROM.
[maxPowerSupportedCharacteristic]	*floating point number*	Maximum slot power in watts.
[mediaType]	*string*	The effective media type, normally thedetectedMediaType, but can be overridden by cli config.
[power]	*InterfacesTransceiverHardwarePower* see 2.506.1.1	Power information.
[powerCharacteristic]	*floating point number*	Maximum module power in watts.
[tuning]	*InterfacesTransceiverHardwareTuning* see 2.506.1.2	Tuning information.
[txDisabled]	*boolean*	Optical transmitter disabled.
[wavelength]	*floating point number*	Wavelength of fiber (nm).

2.506.1.1 InterfacesTransceiverHardwarePower

[configuredRxPower]	*floating point number*	RX power in dBm.
[configuredTxPower]	*floating point number*	TX power in dBm.
[operationalRxAttenuation]	*floating point number*	RX attenuation in dB.
[operationalTxPower]	*floating point number*	TX power in dBm.

2.506.1.2 InterfacesTransceiverHardwareTuning

[computedFrequency]	*floating point number*	Computed frequency in GHz.
[computedRxWavelength]	*floating point number*	Computed RX wavelength in nm.
[computedWavelength]	*floating point number*	Computed wavelength in nm.
[configuredChannel]	*integer*	Configured channel.
[configuredFrequency]	*floating point number*	Configured TX frequency in GHz.
[configuredGrid]	*floating point number*	Configured grid in GHz.
[configuredRxFineFrequency]	*floating point number*	Configured RX fine frequency in GHz.
[configuredRxFrequency]	*floating point number*	Configured RX frequency in GHz.
[gridSpacingCapabilities]	*string*	Grid spacing capabilities in GHz.
[operationalChannel]	*integer*	Operational channel.
[operationalFrequency]	*floating point number*	Operational TX frequency in GHz.
[operationalGrid]	*floating point number*	Operational grid in GHz.
[operationalRxFrequency]	*floating point number*	Operational RX frequency in GHz.
[operationalRxWavelength]	*floating point number*	Operational RX wavelength in nm.

[operationalWavelength]	*floating point number*	Operational TX wavelength in nm.

2.507 show interfaces transceiver performance-monitoring [...]

interfaces	*table(Interface name) = InterfaceTransceiverPerformanceMonitoring* see 2.507.1	Mapping between interface name and the performance monitoring information.
[performanceMonitoringPeriodSeconds]	*integer*	Performance monitoring period in seconds.

2.507.1 InterfaceTransceiverPerformanceMonitoring

[displayName] **	*string*	Interfaces corresponding to the channel of the transceiver. **Since: 4.30.0
[performanceMonitoringIntervals]	*table(numeric string) = InterfaceTransceiverPerformanceMonitoringInterval* see 2.507.1.1	A mapping of performance monitoring interval number to per interval information. Interval numbers are integers starting from 0. Interval 0 is the interval in progress,Interval 1 is the most recently completed interval and so on.

2.507.1.1 InterfaceTransceiverPerformanceMonitoringInterval

[intervalStartTime]	*floating point number*	Time when this interval started in UTC.
[parameters]	*table(string) = InterfaceTransceiverDomParameter* see 2.504.2.1	A mapping of performance monitoring parameter names to per channel value and thresholds in this interval.
[updateTime]	*floating point number*	Last update time in UTC.

2.508 show interfaces transceiver properties [...]

interfaces	*table(Interface name) = TransceiverProperties* see 2.508.1	Mapping between interface name and the transceiver properties.

2.508.1 TransceiverProperties

[adminSpeed]	*TransceiverSpeedLanesDuplex* see 2.508.1.2	Administrative speed/lane count/duplex.
[mediaType]	*string*	Transceiver type.
[operSpeed]	*TransceiverSpeedLanesDuplex* see 2.508.1.2	Operational speed/lane count/duplex.
[xcvrCapabilities]	*list of TransceiverCapabilities* see 2.508.1.1	Transceiver capabilities.

2.508.1.1 TransceiverCapabilities

[fecCapabilities] **	*errorCorrectionCapabilities* see 2.508.1.1.1	Transceiver default host error correction capabilities. **Since: 4.26.0
[speedCapabilities]	*EthSpeedLanesDuplex* see 2.440.1.1.1	Transceiver speed capabilities.

2.508.1.1.1 errorCorrectionCapabilities

[hostErrorCorrection]	*enum: disabled* | *fireCode* | *unsupported* | *reedSolomon*	Transceiver default host error correction.

2.508.1.2 TransceiverSpeedLanesDuplex

[autonegSuffix]	*enum: forced* | *autoNegotiated*	Speed forced or auto-negotiated.
[duplex]	*enum: unconfigured* | *auto* | *half* | *full* | *autoDuplex*	Duplex communication capabilities with this speed.
[laneCount]	*integer*	Number of lanes on the interface.
[speed]	*enum: Unknown* | *200Gbps* | *Hg11Gbps* | *1Gbps* | *100Mbps* | *Hg21Gbps* | *50Gbps* | *25Gbps* | *10Mbps* | *40Gbps* | *Hg27Gbps* | *Hg106Gbps* | *autoSpeed* | *Hg53Gbps* | *auto* | *10Gbps* | *800Gbps* | *100Gbps* | *2p5Gbps* | *Hg42Gbps* | *400Gbps* | *5Gbps*	Interface speed.

2.509 show interfaces transceiver tuning [...]

[interfaces]	*table(Interface name) = InterfaceTransceiverTuning* see 2.509.1	Mapping between interface name and tuning info.

2.509.1 InterfaceTransceiverTuning

rxOutputAmplitude	*InterfaceTransceiverTuningParameter* see 2.509.1.1	RX Output Amplitude.
rxOutputPostEmphasis	*InterfaceTransceiverTuningParameter* see 2.509.1.1	RX Output Post-emphasis.
rxOutputPreEmphasis	*InterfaceTransceiverTuningParameter* see 2.509.1.1	RX Output Pre-emphasis.
txInputEqualization	*InterfaceTransceiverTuningParameter* see 2.509.1.1	TX Input Equalization.

2.509.1.1 InterfaceTransceiverTuningParameter

[detail]	*InterfaceTransceiver-TuningDetail* see 2.509.1.1.1	Detail output of module default and configured values.
[operational]	*string*	The tuning value currently programmed into the module.
supported	*boolean*	This tuning parameter is supported.

2.509.1.1.1 InterfaceTransceiverTuningDetail

[configured]	*string*	The user configured tuning value.
[moduleDefault]	*string*	The default tuning value of the module.

2.510 show interfaces trunk | (trunk (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

trunks	*table(Interface name) = InterfaceTrunk* see 2.510.1	Mapping from an interface name to its' trunk information.

2.510.1 InterfaceTrunk

[activeVlans]	*TrunkVlans* see 2.510.1.1	Vlans allowed and active in management domain.
[allowedVlans]	*TrunkVlans* see 2.510.1.1	Vlans allowed on the trunk. No value means no information is available and array with one element of 0 means no vlans are allowed.
[forwardingVlans]	*TrunkVlans* see 2.510.1.1	Vlans in spannning tree forwarding state.
[linkStatus]	*enum: down | trunking | nonTrunking*	Trunk link status for the interface.
[nativeVlan]	*integer*	Native vlan of the trunk.
[phoneTrunk] **	*boolean*	Mode is 'trunk phone'. **Since: 4.23.1
[portMode]	*enum: dot1qTunnel | tap | routed | trunk | tapTool | tool | access*	Port mode of the trunk.

2.510.1.1 TrunkVlans

[allVlans]	*boolean*	When set it means all vlans 1-4094 in which case the vlanIds will be empty.
vlanIds	*list of integer*	A list of the VLAN Ids.

2.511 show interfaces unidirectional | (unidirectional (module (MODSLOT | (Supervisor MODSLOT_SUPERVISOR) | (Linecard MODSLOT_LINECARD) | (Fabric MODSLOT_FABRIC) | (Switchcard MODSLOT_SWITCHCARD)))) [...]

`interfaces`	*table(Interface name) = InterfaceUnidirectional* see 2.511.1	Mapping between an interface and its unidirectional link information.

2.511.1 InterfaceUnidirectional

`status`	*enum: signal | inactive | connected | unknown | maint-down | errdisabled | disabled | admin | notapplicable | notconnect | noSignal | initializing | uninitialized*	Connection status of the interface.
`uniLinkMode`	*enum: receive-only | disabled | send-receive | send-only | n/a*	Current unidirectional link mode.

2.512 show interfaces vlans [...]

`[interfaces]`	*table(string) = InterfaceVlansModel* see 2.512.1	Map interfaces to VLANs.
`[sourceDetail]`	*string*	Description of the source of the VLAN configs.

2.512.1 InterfaceVlansModel

`[taggedVlans]`	*list of integer*	Tagged VLANs for an interface.
`[untaggedVlan]`	*integer*	Untagged VLAN for an interface.

2.513 show internet-exit policy [...]

policyStatus	*list of SfeIePolicyStatusModel* see 2.513.1	SFE Internet exit policy status.

2.513.1 SfeIePolicyStatusModel

avtId	*integer*	AVT identifier.
avtName	*string*	AVT name.
exitGroups	*list of SfeIeExitGroupInPolicyModel* see 2.513.1.1	Exit groups configured for this policy.
policyName	*string*	Name of the policy.
vniId	*integer*	VNI identifier.
vrfName	*string*	VRF name.

2.513.1.1 SfeIeExitGroupInPolicyModel

exitGroup	*SfeIeExitGroupStatusModel* see 2.513.1.1.1	Exit group model.
isWinning	*boolean*	This is the winning exit group.

2.513.1.1.1 SfeIeExitGroupStatusModel

exitOption	*list of SfeIeExitOptionStatusModel* see 2.513.1.1.1.1	Collection of exit options under this exit group.
groupName	*string*	Name of the exit group.

2.513.1.1.1.1 SfeIeExitOptionStatusModel

[intf]	*string*	Result interface of the connection.
isActive	*boolean*	Exit option is active.
optionKey	*SfeIeExitKeyModel* see 2.513.1.1.1.2	Composite key of the exit option.
[vrf]	*string*	VRF of the result interface.

2.513.1.1.1.2 SfeIeExitKeyModel

name	*string*	Name of the exit.
type	*string*	Type of the exit.

2.514 show inventory

revision 2

[auxiliaryXcvrSlots]	*table(string) = XcvrSlot* see 2.514.8	Dictionary of Auxiliary Transceiver slots indexed by name.
cardSlots	*table(string) = CardSlot* see 2.514.1	Dictionary of Card slots, only present for modular systems.
cpuPortCount	*integer*	Total count of CPU ports.
dataLinkPortCount	*integer*	Total count of data link ports.
fabricPortCount **	*integer*	Total count of Fabric front ports. **Since: 4.30.1
[fabricXcvrSlots] **	*table(string) = XcvrSlot* see 2.514.8	Dictionary of Fabric Transceiver slots indexed by name. **Since: 4.30.1
fanTraySlots	*table(numeric string) = FanTraySlot* see 2.514.3	Dictionary of fan tray slots.
fpgas	*table(string) = Fpga* see 2.514.4	Dictionary of fpgas indexed by fpga name.
internalPortCount	*integer*	Total count of Internal ports.
managementPortCount	*integer*	Total count of Management ports.
mgmtXcvrSlots	*table(string) = XcvrSlot* see 2.514.8	Dictionary of Transceiver slots indexed by name.
portCount	*integer*	Total count of ports.
powerSupplySlots	*table(numeric string) = PowerSupplySlot* see 2.514.5	Dictionary of power supply slots.
[precisionClock]	*Clock* see 2.514.2	System Precision clock.
storageSerials	*table(string) = StorageDevice* see 2.514.6	Dictionary of Storage Devices indexed by Serial Number.
[subcompSerNums]	*table(string) = string*	Dictionary of subcomponent serial numbers.
switchManagementPortCount **	*integer*	Total count of Switch management ports. **Since: 4.29.1
switchedBootstrapPortCount	*integer*	Total count of Switched/Bootstrap ports.
switchedFortyGOnlyPortCount	*integer*	Total count of Switched 40G Only ports.
switchedPortCount	*integer*	Total count of Switched ports.
systemInformation	*SystemInformation* see 2.514.7	System Information.
[unconnectedPortCount]	*integer*	Total count of unconnected ports.
xcvrSlots	*table(string) = XcvrSlot* see 2.514.8	Dictionary of Transceiver slots indexed by name.

2.514.1 CardSlot

hardwareRev	*string*	Hardware version of the card slot.
hwEpoch	*string*	Hardware Epoch.
mfgDate	*string*	Manufacture Date of the card slot (YYYY-MM-DD).

modelName	*string*	Model name of the card slot.
serialNum	*string*	Serial Number of the card slot.

2.514.2 Clock

2.514.3 FanTraySlot

name	*string*	Model name of the fan tray slot.
numFans	*integer*	Number of fans in the fan tray slot.
secondaryFans	*table(numeric string) = string*	Secondary fans, keyed by their slot number.
secondarySerialNum	*string*	Serial number for the fan tray slot on secondary card.
serialNum	*string*	Serial number for the fan tray slot.

2.514.4 Fpga

fpgaRev	*string*	Revision of the FPGA if present.

2.514.5 PowerSupplySlot

name	*string*	Model name of the power supply slot.
serialNum	*string*	Serial number for the power supply slot.

2.514.6 StorageDevice

firmwareRev	*string*	Firmware Revision of the Storage Device.
model	*string*	Model Name of the Storage Device.
mount	*string*	Mount Point of the Storage Device.
serialNum	*string*	Serial Number of the Storage Device.
storageSize	*integer*	Size in GB of the Storage Device.
storageType	*enum: SSD | eUSB | unknown | eMMC | NVME*	Storage Type of the Storage Device.
[supervisor]	*integer*	The supervisor that this device is associated with.

2.514.7 SystemInformation

description	*string*	Model Description.
hardwareRev	*string*	Hardware Revision.
hwEpoch	*string*	Hardware Epoch.
mfgDate	*string*	Manufacture Date (YYYY-MM-DD).
name	*string*	Model name.
serialNum	*string*	Serial Number.

2.514.8 XcvrSlot

hardwareRev	*string*	Transceiver hardware revision number.
mfgName	*string*	Transceiver Manufacturer name.
modelName	*string*	Transceiver model name.
serialNum	*string*	Transceiver serial number.

2.515 show ip

[multicastRouting]	*IpMulticastRouting* see 2.515.1	IPv4 and IPv6 multicast routing information.
v4RoutingEnabled	*boolean*	IPv4 Routing is enabled.
[v4uRpfState]	*enum: disabled* | *errorDisabled* | *enabled*	State of IPv4 uRPF.
[v6EcmpInfo]	*V6EcmpModel* see 2.515.2	IPv6 ECMP Routing Information.
[v6IntfForwarding]	*boolean*	IPv6 interface forwarding is enabled.
v6RoutingEnabled	*boolean*	IPv6 Routing is enabled.
[v6uRpfState]	*enum: disabled* | *errorDisabled* | *enabled*	State of IPv6 uRPF.
[vrrpIntfs]	*integer*	Number of interfaces with VRRP configured.

2.515.1 IpMulticastRouting

[ip6MulticastEnabled]	*boolean*	IPv6 multicast routing is enabled.
[ipMulticastEnabled]	*boolean*	IPv4 multicast routing is enabled.

2.515.2 V6EcmpModel

[nextHopIndex]	*integer*	IPv6 ECMP route nexthop index.
[numPrefixBits]	*integer*	IPv6 ECMP route num prefix bits for nexthop index.
v6EcmpRouteSupport	*boolean*	IPv6 ECMP route is supported.

2.516 show ip access-lists [...]

see 2.53.1

2.517 show ip address virtual source-nat sourceNatAddr

vrfs	*table(string) = IPv4 address*	A mapping of VRF names to their source NAT IP address.

2.518 show ip arp inspection interface [...]

interfaceConfigs	*table(Interface name) = ArpInspectionInterface-Config* see 2.518.1	Interfaces with non-default configurations.

2.518.1 ArpInspectionInterfaceConfig

errdisableBurstInterval	*integer*	Consecutive interval in seconds over which the interface is monitored for disabling.
errdisableRateLimit	*integer*	Maximum number of Arp packets per second before the interface is disabled.
interfaceTrusted	*boolean*	Interface trusted with Arp traffic.
loggingBurstInterval	*integer*	Consecutive interval in seconds over which the interface is monitored for logging.
loggingRateLimit	*integer*	Maximum number of Arp packets per second before a log message is emitted.

2.519 show ip arp inspection statistics vlan | interface [...]

intfStatistics	*table(Interface name) = ArpInspectionStatistics* see 2.519.1	Arp inspection statistics for interfaces.
vlanStatistics	*table(numeric string) = VlanArpInspectionStatistics* see 2.519.3	Arp inspection statistics for vlans.

2.519.1 ArpInspectionStatistics

Could also be an instance of:
IntfArpInspectionStatistics 2.519.2
VlanArpInspectionStatistics 2.519.3

[lastDroppedPacket]	*ArpPacketBrief* see 2.519.1.1	The last Arp packet dropped.
requestsDropped	*integer*	Number of Arp requests dropped.
requestsForwarded	*integer*	Number of Arp requests forwarded.
responsesDropped	*integer*	Number of Arp responses dropped.
responsesForwarded	*integer*	Number of Arp responses forwarded.

2.519.1.1 ArpPacketBrief

arpOpcode	*enum: invalid | response | request*	Arp operation type.
destinationMac	*MAC address*	Destination Mac address.
[dropReason]	*enum: rateLimitExceeded | invalidBinding*	Drop reason of the Arp packet.
receivedInterfaceId	*interface name*	Interface Arp packet was received on.
receivedVlanId	*integer*	Vlan Arp packet was received on.
senderIp	*IPv4 address*	IPv4 address of the sender.
senderMac	*MAC address*	Mac address of the sender.
sourceMac	*MAC address*	Source Mac address.
timestamp	*floating point number*	Time packet was received.

2.519.2 IntfArpInspectionStatistics

intfId	*interface name*	Interface name.
[lastDroppedPacket]	*ArpPacketBrief* see 2.519.1.1	The last Arp packet dropped.
requestsDropped	*integer*	Number of Arp requests dropped.
requestsForwarded	*integer*	Number of Arp requests forwarded.
responsesDropped	*integer*	Number of Arp responses dropped.
responsesForwarded	*integer*	Number of Arp responses forwarded.

2.519.3 VlanArpInspectionStatistics

[lastDroppedPacket]	*ArpPacketBrief* see 2.519.1.1	The last Arp packet dropped.
name	*string*	Vlan name.

requestsDropped	*integer*	Number of Arp requests dropped.
requestsForwarded	*integer*	Number of Arp requests forwarded.
responsesDropped	*integer*	Number of Arp responses dropped.
responsesForwarded	*integer*	Number of Arp responses forwarded.
vlanId	*integer*	Vlan id.

2.520 show ip arp inspection vlan [...]

vlanStatus	*table(numeric string) = VlanArpInspectionStatus* see 2.520.1	VLANs with Arp inspection enabled on the switch.

2.520.1 VlanArpInspectionStatus

configuration	*boolean*	Arp Inspection enabled on this VLAN.
operationState	*boolean*	Arp Inspection operational state on this VLAN.

2.521 show ip as-path access-list NAME detail

activeIpAsPathLists	*table(string) = IpAsPathList* see 2.521.1	Active AS path lists under the current regex mode.
inactiveIpAsPathLists	*table(string) = IpAsPathList* see 2.521.1	Inactive AS path lists under the current regex mode.
regexMode	*enum: string | asn*	Regex mode.

2.521.1 IpAsPathList

ipAsPathEntries	*list of IpAsPathEntry* see 2.521.1.1	AS path entries.
[ipAsPathListSource] **	*string*	Source URL of the as-path access list. **Since: 4.20.0

2.521.1.1 IpAsPathEntry

filterType	*enum: permit | deny*	Filter type.
origin	*enum: bgpIncomplete | bgpEGP | bgpAny | bgpIGP*	Origin of path information.
userRegex	*string*	Regex describing AS path being filtered.
[valid] **	*boolean*	Entry validity given current regex mode for this as-path entry list. **Since: 4.20.5

2.522 show ip as-path access-list NAME summary

[accessListSource]	*string*	Access list source.
[entryCount]	*integer*	Access list entry count.
[sourceTimeStamp]	*floating point number*	Access list source timestamp.

2.523 show ip as-path access-list [...]

see 2.521

2.524 show ip bgp ((community regex COMMREGEX as-path regex ASPREGEX) | (regex ASPREGEX community regex COMMREGEX)) [...]

revision 3

see 2.17

2.525 show ip bgp (IP | PREFIX) [...]

revision 3

see 2.17

2.526 show ip bgp (installed | not-installed) [...]

revision 3

see 2.17

2.527 show ip bgp [...]

revision 3

see 2.17

2.528 show ip bgp access-list AS_PATH_LIST_NAME [...]

revision 3

see 2.17

2.529 show ip bgp community regex REGEX [...]

revision 3

see 2.17

2.530 show ip bgp field-set FSNAME [...]

revision 3

see 2.17

2.531 show ip bgp large-community { LARGE_COMM_VALUE } detail exact [...]

revision 3

see 2.17

2.532 show ip bgp neighbors [...]

revision 2

vrfs	*table(string) = BgpPeerList* see 2.532.1	Per Vrf BGP Peer Information.

2.532.1 BgpPeerList

peerList	*list of BgpPeer* see 2.532.1.1	Bgp Peer List.

2.532.1.1 BgpPeer

[advertisedCommunities] **	*CommunityTypesAdv* see 2.532.1.1.3	Types of communities advertised. **Since: 4.29.0
[advertisedRestartTime]	*integer*	Advertised restart time.
[afiSafiInfo] **	*table(string) = PerAfiSafiInfo* see 2.532.1.1.13	A dictionary of address families. **Since: 4.27.2
[aigpSessionIpv4LabeledUni]	*boolean*	AIGP session for IPv4 Labelled Unicast.
[aigpSessionIpv4Uni]	*boolean*	AIGP session for IPv4 Unicast.
[aigpSessionIpv6LabeledUni]	*boolean*	AIGP session for IPv6 Labelled Unicast.
[aigpSessionIpv6Uni]	*boolean*	AIGP session for IPv6 Unicast.
[ancestorPeerGroups] **	*list of string*	Ancestor peer group list. **Since: 4.29.1
[as4]	*integer*	AS.
asn	*string*	Autonomous system number.
[autoLocalAddress]	*string*	Auto local address.
[bfdDampingMaxTime]	*integer*	BFD flap damping max time.
[bfdDampingTime]	*integer*	BFD current flap damping time.
[bfdDampingTimeLeft]	*floating point number*	BFD flap damping time left.
[bfdMinRx] **	*integer*	BFD receive rate in milliseconds. **Since: 4.28.2
[bfdMinTx] **	*integer*	BFD transmit rate in milliseconds. **Since: 4.28.2
[bfdMult] **	*integer*	BFD multiplier value. **Since: 4.28.2
[bfdSessionState]	*enum: unknown | down | adminDown | init | up*	BFD session state when BFD is enabled.
[bfdState]	*integer*	BFD state.
[bgpPeerCaps]	*integer*	BGP capabilities sent by us.
[bgpPeerFlags]	*list of integer*	BGP peer flag list.
[bgpPeerHiscaps]	*integer*	BGP capabilities received from peer.
[bgpPeerOptions]	*integer*	BGP peer options.
[bgpPeerOptions2]	*integer*	BGP peer options 2.
[bgpPeerOptions3]	*integer*	BGP peer options 3.
bgpSoftReconfigInbound	*enum: All | None | Default*	Which inbound routes that are not accepted to retain for soft reconfig.
[clientReflectionMode]	*enum: meshed | clientToClientDisabled*	Route reflector mode.

[clusterId] **	*IPv4 address*	Cluster ID in use by this peer. **Since: 4.29.0
[confHoldTime]	*integer*	Configured hold time.
[confRemotePort]	*integer*	Configured remote port.
[configHoldTime]	*integer*	Configured hold time (multi-agent mode).
[configIdleHoldTime]	*integer*	Configured idle hold time.
[configKeepaliveTime]	*integer*	Configured keepalive time (multi-agent mode).
[configMinHoldTime] **	*integer*	Configured minimum hold time in seconds. **Since: 4.29.0
[configRestartTime]	*integer*	Configured restart time (multi-agent mode).
[configuredKeepaliveTime]	*integer*	Configured keepalive time.
[connectFailed]	*integer*	Failed connection attempts.
[connectInterval]	*integer*	Connection interval.
[connectTimeLeft]	*integer*	Connection time left.
[connectTimerActive]	*boolean*	Connect timer is active.
[description]	*string*	Description.
[dpsBestEcmpPaths]	*integer*	Dynamic Path Selection best ECMP paths received.
[dpsBestPaths]	*integer*	Dynamic Path Selection best paths received.
[dpsPrefixesReceived]	*integer*	Dynamic Path Selection prefixes received.
[dpsPrefixesSent]	*integer*	Dynamic Path Selection prefixes sent.
[dropStats]	*PeerDropStats* see 2.532.1.1.12	Prefix drop statistics.
[enRRefreshstalePath-Removal] **	*boolean*	Enhanced route refresh stale path removal enabled. **Since: 4.30.0
[endOfRibTimeLeft]	*integer*	End of rib time left.
[endOfRibs] **	*EndOfRibs* see 2.532.1.1.4	EOR Information for all negotiated AfiSafi(s). **Since: 4.28.2
[eorTimerActive]	*boolean*	End-of-Rib timer is active.
[establishFailHint]	*string*	Establish failure hint.
[establishedTime]	*integer*	Established time.
[establishedTransitions]	*integer*	Established transitions.
[evpnBestEcmpPaths]	*integer*	Evpn best ECMP paths received.
[evpnBestPaths]	*integer*	Evpn best paths received.
[evpnReceived]	*integer*	Ethernet VPN prefixes received.
[evpnSent]	*integer*	Ethernet VPN prefixes sent.
[fwdFailoverTriggerSession-Ipv4Uni] **	*boolean*	Forwarding Failover Trigger Session for IPv4 Unicast. **Since: 4.28.2
[fwdFailoverTriggerSession-Ipv6Uni] **	*boolean*	Forwarding Failover Trigger Session for IPv6 Unicast. **Since: 4.28.2
[grRestartTimeLeft]	*integer*	Restart time left (multi-agent mode).

[grStalePolicies] **	*GrHelperStalePolicies* see 2.532.1.1.6	Graceful Restart stale policy information per Afi/Safi. **Since: 4.30.0
[gracefulRestartTimer-Active]	*boolean*	Graceful Restart timer is active.
[holdTime]	*integer*	Hold time.
[holdTimeLeft]	*integer*	Hold time left.
[idleHoldTimeLeft]	*integer*	Idle hold time left.
[idleReason]	*string*	Idle reason if session state is idle.
[idleRestartTimerActive]	*boolean*	Idle restart timer active.
[ifName]	*string*	Interface name.
[inMessageStats]	*MessageStatInfo* see 2.532.1.1.8	Input message statistics.
[invalidNextHopDirectEbgp]	*boolean*	Nexthop invalid for single hop EBGP.
[keepaliveTime]	*integer*	Keepalive time.
[keepaliveTimeLeft]	*integer*	Keepalive time left.
[lastErrorCode]	*string*	Last send error code.
[lastErrorData]	*string*	Last send error data.
[lastErrorRcvdCode]	*string*	Last receive error code.
[lastErrorRcvdData]	*string*	Last receive error data.
[lastErrorRcvdSubcode]	*string*	Last receive error subcode.
[lastErrorRcvdTimeInfo]	*ErrorTimeInfo* see 2.532.1.1.5	Last receive error time information.
[lastErrorSubcode]	*string*	Last send error subcode.
[lastErrorTimeInfo]	*ErrorTimeInfo* see 2.532.1.1.5	Last send error time information.
[lastEvent]	*string*	Last BGP event.
[lastRcvd]	*integer*	Last received.
[lastSent]	*integer*	Last sent.
[lastSocketInErrorMsg]	*string*	Last socket in error message.
[lastSocketInErrorTimeInfo]	*ErrorTimeInfo* see 2.532.1.1.5	Last socket in error time information.
[lastSocketOutErrorMsg]	*string*	Last socket out error message.
[lastSocketOutErrorTime-Info]	*ErrorTimeInfo* see 2.532.1.1.5	Last socket out error time information.
[lastState]	*string*	Last BGP state.
[linkStateBestEcmpPaths]	*integer*	Link State best ECMP paths.
[linkStateBestPaths]	*integer*	Link State best paths received.
[linkStateReceived]	*integer*	Link State NLRIs received.
[linkStateSent]	*integer*	Link State NLRIs sent.
linkType	*string*	Link type.
[lldpNeighbors] **	*list of LldpNeighbor* see 2.532.1.1.7	LLDP Neighbors list. **Since: 4.27.2
[llgrTimers] **	*table(string) = integer*	Long lived graceful restart stale path timers per received Afi/Safi. **Since: 4.29.0
[localAddressPort]	*integer*	Local address port.
[localAddressV4]	*IP address*	Local IPv4 address.
[localAddressV6]	*IP address*	Local IPv6 address.
localAsn	*string*	Local autonomous system number.
[localPort]	*integer*	Local port.

localRouterId	*IPv4 address*	Local router ID.
[maintenance]	*boolean*	Peering under maintenance.
[maintenanceRm]	*string*	Maintenance mode route map.
[maintenanceRmIn]	*string*	Maintenance mode inbound route map.
[maintenanceRmOut]	*string*	Maintenance mode outbound route map.
[maxTotalAcceptedRoutes] **	*integer*	Maximum number of accepted routes (0 means unlimited). **Since: 4.28.1
[maxTotalRoutes] **	*integer*	Maximum number of received routes (0 means unlimited). **Since: 4.28.1
maxTtlHops	*integer*	Maximum hops allowed for BGP neighbor.
[md5AuthEnabled]	*boolean*	MD5 authentication is enabled.
[minHoldTime] **	*integer*	Minimum hold time in seconds. **Since: 4.29.0
[miscFlags]	*integer*	BGP miscellaneous flags.
[missingPolicy]	*MissingPolicyTypes* see 2.532.1.1.10	Import/Export missing policies per AFI/SAFI.
[missingPolicyDefaultDeny]	*MissingPolicyDefaultDeny* see 2.532.1.1.9	Missing policy/default deny status.
negotiatedVersion	*integer*	Negotiated BGP version.
[neighborCapabilities]	*NeighborCapabilities* see 2.532.1.1.11	BGP neighbor capabilities.
[nexthopLuOriginateIpv4Uni]	*boolean*	Next hop labeled unicast origination for IPv4 Unicast.
[nexthopLuOriginateIpv6Uni]	*boolean*	Next hop labeled unicast origination for IPv6 Unicast.
[nexthopLuOriginateLfib-BackupIpForwardIpv4Uni]	*boolean*	Next hop labeled unicast origination LFIB backup IP forwarding for IPv4 Unicast.
[nexthopLuOriginateLfib-BackupIpForwardIpv6Uni]	*boolean*	Next hop labeled unicast origination LFIB backup IP forwarding for IPv6 Unicast.
[noComms]	*boolean*	Send no communities.
[noComs]	*boolean*	Send no communites(multi-agent mode).
[orrPosition]	*table(string) = BgpOrrPosition* see 2.190.1	BGP optimal route reflection position per address-family.
[outMessageStats]	*MessageStatInfo* see 2.532.1.1.8	Output message statistics.
[outboundEnRRefreshEnabled] **	*boolean*	Outbound enhanced route refresh enabled. **Since: 4.30.0
[passiveConnEnabled]	*boolean*	Passive TCP connection mode is enabled.
peerAddress	*string*	Peer address.
[peerFilter]	*string*	Peer filter name.
[peerGroup]	*string*	Peer group name.
[peerGroupName]	*string*	Peer group name (multi-agent mode).
[peerInUpdateErrors]	*BgpPeerInUpdateErrors* see 2.532.1.1.1	Peer inbound update errors.
[peerTcpInfo]	*BgpPeerTcpInfo* see 2.532.1.1.2	Peer TCP information.
[prefixListInfo]	*RouteFilterInfo* see 2.532.1.1.14	Neighbor prefix list information.

[prefixesReceived]	*integer*	IPv4 unicast prefixes received.
[prefixesSent]	*integer*	IPv4 unicast prefixes sent.
[prependOwnDisabled]	*boolean*	Do not prepend own AS number.
[rcfInbound]	*string*	Rcf inbound.
[rcfInfo]	*RouteFilterInfo* see 2.532.1.1.14	Neighbor RCF information (supersedes route map).
[rcfOutbound]	*string*	Rcf outbound.
[receivedMessages]	*integer*	Received messages.
[receivedUpdates]	*integer*	Received updates.
[remoteAddrPort]	*integer*	Remote address port.
[remoteAddressV4]	*IP address*	Remote IPv4 address.
[remoteAddressV6]	*IP address*	Remote IPv6 address.
[remotePort]	*integer*	Remote port.
[removeOrReplacePrivateAs-Egress]	*enum: removePrivateAsEgress* | *removePrivateAsEgress-Replace* | *removePrivateAsEgressAll*	From outbound updates to this neighbor: removePrivateAsEgress - Remove private AS numbers when there are only private AS numbers in inbound/outbound AS path lists. removePrivateAsEgressAll - Always remove private AS numbers in outbound AS path lists. removePrivateAsEgressReplace - Replace private AS numbers with the local AS number in outbound AS path lists.
[removeOrReplacePrivateAs-Ingress]	*enum: removePrivateAsIngress* | *removePrivateAsIngress-Replace*	From inbound updates to this neighbor: removePrivateAsIngress - Always remove private AS numbers in inbound AS path lists. removePrivateAsIngressReplace - Replace private AS numbers with the local AS number in inbound AS path lists.
[restart]	*integer*	Received restart time.
[restartTimeLeft]	*integer*	Restart time left.
[routeMapInbound]	*string*	Route map inbound (overridden if RCF is configured).
[routeMapInfo]	*RouteMapFilterInfo* see 2.532.1.1.15	Neighbor route map information.
[routeMapOutbound]	*string*	Route map outbound (overridden if RCF is configured).
[routeReflectorClient]	*boolean*	Neighbor is a route reflector client.
routerId	*IPv4 address*	BGP router identity.
[rpkiOriginValidation-Method]	*enum: disabled* | *preferCommunity* | *community* | *local*	Method used for validating prefix origin AS.
[rpkiOriginValidationRoute-Map]	*string*	Route map to filter which prefixes should be validated.
[rpkiOriginValidationSend-ExtComm]	*boolean*	Attach an extended community carrying the prefix origin validity.
[rtMembershipBestEcmpPaths]	*integer*	RT Membership best ECMP paths.
[rtMembershipBestPaths]	*integer*	RT Membership best paths.
[rtMembershipPrefixes-Received]	*integer*	RT Membership prefixes received.
[rtMembershipPrefixesSent]	*integer*	RT Membership prefixes sent.
[sentMessages]	*integer*	Sent messages.

[sentUpdates]	*integer*	Sent updates.
[stalePathRemovalTimer]	*table(string) = integer*	Inbound enhanced route refresh stale path removal timer per negotiated Afi/Safi.
[state]	*string*	BGP state.
[tcpAoInfo] **	*TcpAoInfo* see 2.532.1.1.16	TCP-AO information. **Since: 4.28.1
[thirdPartyNexthopLu-OriginateIpv4Uni]	*boolean*	Next hop labeled unicast origination for received IPv4 Unicast nexthops.
[thirdPartyNexthopLu-OriginateIpv6Uni]	*boolean*	Next hop labeled unicast origination for received IPv6 Unicast nexthops.
[totalAcceptedRoutesWarn-Limit] **	*integer*	Number of accepted routes after which a warning is issued (0 means never warn). **Since: 4.28.1
[totalRoutesWarnLimit] **	*integer*	Number of received routes after which a warning is issued (0 means never warn). **Since: 4.28.1
ttl	*integer*	TTL value for peer connections.
[ttlConfig]	*enum: ebgpMultiHop | gtsm*	TTL max hops configuration mode.
updateGroupIndex	*integer*	Update group index.
[updateSource]	*IP address*	Local TCP address.
[updownTime]	*integer*	Up down time.
[usesGlobalClusterId] **	*boolean*	This peer is using the global cluster ID. **Since: 4.29.0
[v4BestEcmpPaths]	*integer*	IPv4 best ECMP paths received.
[v4BestPaths]	*integer*	IPv4 best paths received.
[v4FlowspecBestEcmpPaths]	*integer*	IPv4 Flow Specification best ECMP paths received.
[v4FlowspecBestPaths]	*integer*	IPv4 Flow Specification best paths received.
[v4FlowspecReceived]	*integer*	IPv4 Flow Specification rules received.
[v4FlowspecSent]	*integer*	IPv4 Flow Specification rules sent.
[v4FlowspecVpnBestEcmp-Paths]	*integer*	VPN-IPv4 Flow Specification best ECMP paths received.
[v4FlowspecVpnBestPaths]	*integer*	VPN-IPv4 Flow Specification best paths received.
[v4FlowspecVpnReceived]	*integer*	VPN-IPv4 Flow Specification rules received.
[v4FlowspecVpnSent]	*integer*	VPN-IPv4 Flow Specification rules sent.
[v4LuBestEcmpPaths]	*integer*	IPv4 LU best ECMP paths received.
[v4LuBestPaths]	*integer*	IPv4 LU best paths received.
[v4LuPrefixesReceived]	*integer*	IPv4 LU prefixes received.
[v4LuPrefixesSent]	*integer*	IPv4 LU prefixes sent.
[v4MappedV6NextHop-TranslationIPv6LabeledUni] **	*boolean*	Next hop resolution translates IPv4 mapped IPv6 nexthop addresses to IPv4 addresses for IPv6 with MPLS Labels. **Since: 4.27.2
[v4MplsVpnBestEcmpPaths]	*integer*	IPv4 MPLS VPN best ECMP paths received.
[v4MplsVpnBestPaths]	*integer*	IPv4 MPLS VPN best paths received.
[v4MplsVpnPrefixesReceived]	*integer*	IPv4 MPLS VPN prefixes received.
[v4MplsVpnPrefixesSent]	*integer*	IPv4 MPLS VPN prefixes sent.

[v4MulticastBestEcmpPaths]	*integer*	IPv4 Multicast best ECMP paths received.
[v4MulticastBestPaths]	*integer*	IPv4 Multicast best paths received.
[v4MulticastReceived]	*integer*	IPv4 Multicast prefixes received.
[v4MulticastSent]	*integer*	IPv4 Multicast prefixes sent.
[v4RfdMaxPathsSuppressed-Last24Hours] **	*integer*	Peak IPv4 unicast paths suppressed due to route flap damping in the last 24 hours. **Since: **currentRelease**
[v4RfdPathSuppressions-Last24Hours] **	*integer*	IPv4 unicast path suppression events due to route flap damping in the last 24 hours. **Since: **currentRelease**
[v4RfdPathsSuppressed] **	*integer*	IPv4 unicast paths currently suppressed due to route flap damping. **Since: **currentRelease**
[v4RfdSuppressionsSince-Tracking] **	*integer*	Lifetime IPv4 unicast path suppression events due to route flap damping. **Since: **currentRelease**
[v4SrTeBestEcmpPaths]	*integer*	IPv4 segment routing traffic engineering best ECMP paths received.
[v4SrTeBestPaths]	*integer*	IPv4 segment routing traffic engineering best paths received.
[v4SrTePrefixesReceived]	*integer*	IPv4 segment routing traffic engineering policies received.
[v4SrTePrefixesSent]	*integer*	IPv4 segment routing traffic engineering policies sent.
[v4gateway]	*IP address*	IPv4 gateway address.
[v6BestEcmpPaths]	*integer*	IPv6 best ECMP paths received.
[v6BestPaths]	*integer*	IPv6 best paths received.
[v6FlowspecBestEcmpPaths]	*integer*	IPv6 Flow Specification best ECMP paths received.
[v6FlowspecBestPaths]	*integer*	IPv6 Flow Specification best paths received.
[v6FlowspecReceived]	*integer*	IPv6 Flow Specification rules received.
[v6FlowspecSent]	*integer*	IPv6 Flow Specification rules sent.
[v6FlowspecVpnBestEcmp-Paths]	*integer*	VPN-IPv6 Flow Specification best ECMP paths received.
[v6FlowspecVpnBestPaths]	*integer*	VPN-IPv6 Flow Specification best paths received.
[v6FlowspecVpnReceived]	*integer*	VPN-IPv6 Flow Specification rules received.
[v6FlowspecVpnSent]	*integer*	VPN-IPv6 Flow Specification rules sent.
[v6LuBestEcmpPaths]	*integer*	IPv6 LU best ECMP paths received.
[v6LuBestPaths]	*integer*	IPv6 LU best paths received.
[v6LuPrefixesReceived]	*integer*	IPv6 LU prefixes received.
[v6LuPrefixesSent]	*integer*	IPv6 LU prefixes sent.
[v6MplsVpnBestEcmpPaths]	*integer*	IPv6 MPLS VPN best ECMP paths received.
[v6MplsVpnBestPaths]	*integer*	IPv6 MPLS VPN best paths received.
[v6MplsVpnPrefixesReceived]	*integer*	IPv6 MPLS VPN prefixes received.
[v6MplsVpnPrefixesSent]	*integer*	IPv6 MPLS VPN prefixes sent.
[v6MulticastBestEcmpPaths]	*integer*	IPv6 Multicast best ECMP paths received.
[v6MulticastBestPaths]	*integer*	IPv6 Multicast best paths received.

[v6MulticastReceived]	*integer*	IPv6 Multicast prefixes received.
[v6MulticastSent]	*integer*	IPv6 Multicast prefixes sent.
[v6PrefixesReceived]	*integer*	IPv6 Unicast prefixes received.
[v6PrefixesSent]	*integer*	IPv6 Unicast prefixes sent.
[v6RfdMaxPathsSuppressed-Last24Hours] **	*integer*	Peak IPv6 unicast paths suppressed due to route flap damping in the last 24 hours. **Since: **currentRelease**
[v6RfdPathSuppressions-Last24Hours] **	*integer*	IPv6 unicast path suppression events due to route flap damping in the last 24 hours. **Since: **currentRelease**
[v6RfdPathsSuppressed] **	*integer*	IPv6 unicast paths currently suppressed due to route flap damping. **Since: **currentRelease**
[v6RfdSuppressionsSince-Tracking] **	*integer*	Lifetime IPv6 unicast path suppression events due to route flap damping. **Since: **currentRelease**
[v6SixPeBestEcmpPaths]	*integer*	IPv6 provider edge (6PE) best ECMP paths received.
[v6SixPeBestPaths]	*integer*	IPv6 provider edge (6PE) best paths received.
[v6SixPePrefixesReceived]	*integer*	IPv6 provider edge (6PE) prefixes received.
[v6SixPePrefixesSent]	*integer*	IPv6 provider edge (6PE) prefixes sent.
[v6SrTeBestEcmpPaths]	*integer*	IPv6 segment routing traffic engineering best ECMP paths received.
[v6SrTeBestPaths]	*integer*	IPv6 segment routing traffic engineering best paths received.
[v6SrTePrefixesReceived]	*integer*	IPv6 segment routing traffic engineering policies received.
[v6SrTePrefixesSent]	*integer*	IPv6 segment routing traffic engineering policies sent.
[v6gateway]	*IP address*	IPv6 gateway address.
version	*integer*	BGP version.
[vplsBestEcmpPaths] **	*integer*	VPLS best ECMP paths received. **Since: 4.28.1
[vplsBestPaths] **	*integer*	VPLS best paths received. **Since: 4.28.1
[vplsReceived] **	*integer*	VPLS prefixes received. **Since: 4.28.1
[vplsSent] **	*integer*	VPLS prefixes sent. **Since: 4.28.1
vrf	*string*	Vrf.

2.532.1.1.1 BgpPeerInUpdateErrors

[disableAfiSafiAttr]	*string*	last attribute resulting in AFI/SAFI disable.
[disabledAfiSafi]	*string*	disabled AFI/SAFI.
[ignoreAttr]	*string*	last attribute resulting in ignore.
[inUpdErrDisableAfiSafi]	*integer*	in updates disabling AFI/SAFI.
[inUpdErrIgnore]	*integer*	in updates ignored attributes.
[inUpdErrWithdraw]	*integer*	in updates treated as withdraw.

[lastUpdErrTime]	*integer*	last time of Update inbound error.
[withdrawAttr]	*string*	last attribute resulting in withdraw.

2.532.1.1.2 BgpPeerTcpInfo

congestionWindow	*integer*	TCP send congestion window.
[connectionOptions]	*TcpConnOptions* see 2.532.1.1.2.1	TCP socket optional settings.
delayedAckTimeout	*integer*	TCP socket delayed ACK timeout (microseconds).
inputMaxQueueLength	*integer*	TCP input queue max length.
inputQueueLength	*integer*	TCP input queue length.
maxSegmentSize	*integer*	TCP outgoing maximum segment size.
options	*integer*	TCP socket options.
outputMaxQueueLength	*integer*	TCP output queue max length.
outputQueueLength	*integer*	TCP output queue length.
rcvRtt	*integer*	TCP receive round-trip time (microseconds).
rcvWindow	*integer*	TCP advertised receive window.
rcvWindowScale	*integer*	TCP socket receive window scale.
retransTimeout	*integer*	TCP socket retransmission timeout (microseconds).
sendWindowScale	*integer*	TCP socket send window scale.
slowStartThreshold	*integer*	TCP send slow start size threshold.
sndRtt	*integer*	TCP round-trip time (microseconds).
sndRttVariance	*integer*	TCP round-trip time variance (microseconds).
state	*enum: CLOSING* | *NOTUSED* | *LISTEN* | *SYN-RECEIVED* | *CLOSE-WAIT* | *FIN-WAIT-2* | *CLOSED* | *LAST-ACK* | *ESTABLISHED* | *TIME-WAIT* | *FIN-WAIT-1* | *SYN-SENT*	TCP socket state.
totalRetrans	*integer*	Total number of TCP retransmissions.

2.532.1.1.2.1 TcpConnOptions

ecnEnabled	*boolean*	Explicit Congestion Notification (ECN) is enabled.
selectiveAcknowledgements-Enabled	*boolean*	Selective Acknowledgments are enabled.
timestampsEnabled	*boolean*	Timestamps are enabled.
windowScaleEnabled	*boolean*	Window Scale is enabled.

2.532.1.1.3 CommunityTypesAdv

extended	*boolean*	Extended community advertised.
large	*boolean*	Large community advertised.
standard	*boolean*	Standard community advertised.

2.532.1.1.4 EndOfRibs

[incoming]	*EndOfRib* see 2.532.1.1.4.1	Incoming EOR Information.

2.532.1.1.4.1 EndOfRib

[dps]	*EndOfRibInfo* see 2.532.1.1.4.2	Dynamic Path Selection EOR Information.
[ipv4FlowSpec]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 flow specification EOR Information.
[ipv4FlowSpecVpn] **	*EndOfRibInfo* see 2.532.1.1.4.2	VPN-IPv4 flow specification EOR Information. **Since: 4.30.0
[ipv4MplsLabels]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 labeled-unicast EOR Information.
[ipv4MplsVpn]	*EndOfRibInfo* see 2.532.1.1.4.2	VPN-IPv4 EOR Information.
[ipv4Multicast]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 multicast EOR Information.
[ipv4Mvpn]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 MVPN EOR Information.
[ipv4SrTe]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 SR-TE EOR Information.
[ipv4Unicast]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv4 unicast EOR Information.
[ipv6FlowSpec]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv6 flow specification EOR Information.
[ipv6FlowSpecVpn] **	*EndOfRibInfo* see 2.532.1.1.4.2	VPN-IPv6 flow specification EOR Information. **Since: 4.30.0
[ipv6MplsLabels]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv6 labeled-unicast EOR Information.
[ipv6MplsVpn]	*EndOfRibInfo* see 2.532.1.1.4.2	VPN-IPv6 EOR Information.
[ipv6Multicast]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv6 multicast EOR Information.
[ipv6SrTe]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv6 SR-TE EOR Information.
[ipv6Unicast]	*EndOfRibInfo* see 2.532.1.1.4.2	IPv6 unicast EOR Information.
[l2VpnEvpn]	*EndOfRibInfo* see 2.532.1.1.4.2	L2VPN EVPN EOR Information.
[l2VpnVpls]	*EndOfRibInfo* see 2.532.1.1.4.2	L2VPN VPLS EOR Information.
[linkState]	*EndOfRibInfo* see 2.532.1.1.4.2	BGP Link State EOR Information.

[rtMembership]	*EndOfRibInfo* see 2.532.1.1.4.2	RT Membership EOR Information.

2.532.1.1.4.2 EndOfRibInfo

[eorReceivedTime]	*floating point number*	UTC time at which EOR was received.
[stalePathsDeleted]	*integer*	Number of stale paths deleted after graceful restart.
status	*enum: timeout | received | pending*	Status of the EOR.

2.532.1.1.5 ErrorTimeInfo

[firstTime]	*integer*	Time at which last error occured for the first time.
[repeats]	*integer*	Number of times last error repeated.
[time]	*integer*	Time at which last error occured.

2.532.1.1.6 GrHelperStalePolicies

[ipv4MplsLabels]	*GrHelperStalePolicy* see 2.532.1.1.6.1	IPv4 Labeled Unicast stale policy information.
[ipv4Unicast]	*GrHelperStalePolicy* see 2.532.1.1.6.1	IPv4 Unicast stale policy information.
[ipv6MplsLabels]	*GrHelperStalePolicy* see 2.532.1.1.6.1	IPv6 Labeled Unicast stale policy information.
[ipv6Unicast]	*GrHelperStalePolicy* see 2.532.1.1.6.1	IPv6 Unicast stale policy information.

2.532.1.1.6.1 GrHelperStalePolicy

[grOptional]	*boolean*	Graceful Restart negotiation is optional.
[staleRmName]	*string*	AFI/SAFI stale policy route map.

2.532.1.1.7 LldpNeighbor

[description]	*string*	LLDP Neighbor system description.
device	*string*	LLDP Neighbor system name.

2.532.1.1.8 MessageStatInfo

[beginOfRtRefreshes]	*integer*	Beginning of Route Refreshes.
[endOfRtRefreshes]	*integer*	End of Route Refreshes.
[keepalives]	*integer*	Keepalives.
[notifications]	*integer*	Notifications.
[opens]	*integer*	Opens.
[queueDepth]	*integer*	Queue Depth.
[rtRefreshes]	*integer*	Route Refreshes.

[updates]	*integer*	Updates.

2.532.1.1.9 MissingPolicyDefaultDeny

[exportActive]	*boolean*	Missing policy/default deny export action is active.
[importActive]	*boolean*	Missing policy/default deny import action is active.

2.532.1.1.10 MissingPolicyTypes

[exportPolicy]	*MissingPolicy* see 2.532.1.1.10.1	AFI/SAFI import missing policy actions.
[importPolicy]	*MissingPolicy* see 2.532.1.1.10.1	AFI/SAFI export missing policy actions.

2.532.1.1.10.1 MissingPolicy

[dps]	*MissingPolicyState* see 2.532.1.1.10.2	Dynamic Path Selection missing policy action.
[ipv4FlowSpec]	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 flow specification missing policy action.
[ipv4FlowSpecVpn] **	*MissingPolicyState* see 2.532.1.1.10.2	VPN-IPv4 flow specification missing policy action. **Since: 4.30.0
[ipv4MplsLabels]	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 labeled-unicast missing policy action.
[ipv4MplsVpn]	*MissingPolicyState* see 2.532.1.1.10.2	VPN-IPv4 missing policy action.
[ipv4Multicast]	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 multicast missing policy action.
[ipv4Mvpn] **	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 MVPN missing policy action. **Since: 4.28.2
[ipv4SrTe]	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 SR-TE missing policy action.
[ipv4Unicast]	*MissingPolicyState* see 2.532.1.1.10.2	IPv4 unicast missing policy action.
[ipv6FlowSpec]	*MissingPolicyState* see 2.532.1.1.10.2	IPv6 flow specification missing policy action.
[ipv6FlowSpecVpn] **	*MissingPolicyState* see 2.532.1.1.10.2	VPN-IPv6 flow specification missing policy action. **Since: 4.30.0
[ipv6MplsLabels]	*MissingPolicyState* see 2.532.1.1.10.2	IPv6 labeled-unicast missing policy action.
[ipv6MplsVpn]	*MissingPolicyState* see 2.532.1.1.10.2	VPN-IPv6 missing policy action.
[ipv6Multicast]	*MissingPolicyState* see 2.532.1.1.10.2	IPv6 multicast missing policy action.
[ipv6SrTe]	*MissingPolicyState* see 2.532.1.1.10.2	IPv6 SR-TE missing policy action.

[ipv6Unicast]	*MissingPolicyState* see 2.532.1.1.10.2	IPv6 unicast missing policy action.
[l2VpnEvpn]	*MissingPolicyState* see 2.532.1.1.10.2	L2VPN EVPN missing policy action.
[l2VpnVpls] **	*MissingPolicyState* see 2.532.1.1.10.2	L2VPN VPLS missing policy action. **Since: 4.28.2
[linkState]	*MissingPolicyState* see 2.532.1.1.10.2	BGP Link State missing policy action.
[rtMembership]	*MissingPolicyState* see 2.532.1.1.10.2	RT Membership missing policy action.

2.532.1.1.10.2 MissingPolicyState

action	*enum: permit | deny*	Missing policy action.
status	*enum: inboundRouteMap-Misconfigured | permit | noRouteMapConfigured | routeMapMisconfiguredBy-OtherAf | misconfiguredPrefixList | outboundRouteMap-Misconfigured | unspecified | namedRouteMapMissing | misconfiguredSubRouteMap*	Missing policy status.

2.532.1.1.11 NeighborCapabilities

[addPathsCaps]	*AddPathsCapabilities* see 2.532.1.1.11.1	Additional paths capabilities.
[dynamicCap]	*Capability* see 2.532.1.1.11.5	Dynamic capabilities.
[enhancedRouteRefreshCap]	*Capability* see 2.532.1.1.11.5	Enhanced Route Refresh capability.
[extendedNextHopCaps]	*ExtendedNexthop-Capabilities* see 2.532.1.1.11.6	Extended Next-Hop capabilities.
[fourOctetAsnCap]	*Capability* see 2.532.1.1.11.5	Four Octet ASN capability.
[grCaps]	*GrCapabilities* see 2.532.1.1.11.7	Graceful Restart capabilities.
[longlivedGrCaps]	*LlgrCapabilities* see 2.532.1.1.11.10	Long Lived Graceful Restart capabilities.
[multipleLabelsCap]	*MultipleLabelsCapabilities* see 2.532.1.1.11.13	Multiple labels capability.
[multiprotocolCaps]	*MultiprotocolCapabilities* see 2.532.1.1.11.15	Multiprotocol capabilities.
[negotiationDisabled]	*boolean*	Neighbor capability negotiation is disabled.
[routeRefreshCap]	*Capability* see 2.532.1.1.11.5	Route Refresh capability.
[sendEorMessages]	*Capability* see 2.532.1.1.11.5	Send End-of-RIB messages.

[unsupportedCapReceived]	*boolean*	Neighbor does not support all required capabilities.

2.532.1.1.11.1 AddPathsCapabilities

[dps]	*AddPathsCapability* see 2.532.1.1.11.2	Dynamic Path Selection additional paths capability.
[ipv4FlowSpec]	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 flow specification additional paths capability.
[ipv4FlowSpecVpn] **	*AddPathsCapability* see 2.532.1.1.11.2	VPN-IPv4 flow specification additional paths capability. **Since: 4.30.0
[ipv4MplsLabels]	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 labeled-unicast additional paths capability.
[ipv4MplsVpn]	*AddPathsCapability* see 2.532.1.1.11.2	VPN-IPv4 additional paths capability.
[ipv4Multicast]	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 multicast additional paths capability.
[ipv4Mvpn] **	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 MVPN additional paths capability. **Since: 4.28.2
[ipv4SrTe]	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 SR-TE additional paths capability.
[ipv4Unicast]	*AddPathsCapability* see 2.532.1.1.11.2	IPv4 unicast additional paths capability.
[ipv6FlowSpec]	*AddPathsCapability* see 2.532.1.1.11.2	IPv6 flow specification additional paths capability.
[ipv6FlowSpecVpn] **	*AddPathsCapability* see 2.532.1.1.11.2	VPN-IPv6 flow specification additional paths capability. **Since: 4.30.0
[ipv6MplsLabels]	*AddPathsCapability* see 2.532.1.1.11.2	IPv6 labeled-unicast additional paths capability.
[ipv6MplsVpn]	*AddPathsCapability* see 2.532.1.1.11.2	VPN-IPv6 additional paths capability.
[ipv6Multicast]	*AddPathsCapability* see 2.532.1.1.11.2	IPv6 multicast additional paths capability.
[ipv6SrTe]	*AddPathsCapability* see 2.532.1.1.11.2	IPv6 SR-TE additional paths capability.
[ipv6Unicast]	*AddPathsCapability* see 2.532.1.1.11.2	IPv6 unicast additional paths capability.
[l2VpnEvpn]	*AddPathsCapability* see 2.532.1.1.11.2	L2VPN EVPN additional paths capability.
[l2VpnVpls] **	*AddPathsCapability* see 2.532.1.1.11.2	L2VPN VPLS additional paths capability. **Since: 4.28.1
[linkState]	*AddPathsCapability* see 2.532.1.1.11.2	BGP Link State additional paths capability.
[rtMembership]	*AddPathsCapability* see 2.532.1.1.11.2	RT Membership additional paths capability.

2.532.1.1.11.2 AddPathsCapability

[receivingAddPaths]	*ReceivingAddPaths* see 2.532.1.1.11.3	Negotiation state of receiving additional paths from peer.

[sendingAddPaths]	*SendingAddPaths* see 2.532.1.1.11.4	Negotiation state of sending additional paths to peer.

2.532.1.1.11.3 ReceivingAddPaths

enabled	*boolean*	Receipt of additional paths from peer is enabled.
receiveCapabilityAdvertised	*boolean*	Advertised capability to receive additional paths from peer.
sendCapabilityReceived	*boolean*	Received capability to send additional paths from peer.

2.532.1.1.11.4 SendingAddPaths

enabled	*boolean*	Sending of additional paths to peer is enabled.
receiveCapabilityReceived	*boolean*	Received capability to receive additional paths from peer.
sendCapabilityAdvertised	*boolean*	Advertised capability to send additional paths to peer.

2.532.1.1.11.5 Capability

advertised	*boolean*	Capability advertised to peer.
enabled	*boolean*	Capability is enabled.
received	*boolean*	Capability received from peer.

2.532.1.1.11.6 ExtendedNexthopCapabilities

[ipv4LocalNlriOverIpv6]	*boolean*	Local IPv4 NLRIs advertised over an IPv6 backbone.
[ipv4UnicastOverIpv6]	*Capability* see 2.532.1.1.11.5	IPv4 Unicast over an IPv6 backbone capability.
[v4MplsVpnOverIpv6] **	*Capability* see 2.532.1.1.11.5	IPv4 MPLS VPN over an IPv6 backbone capability. **Since: 4.30.0

2.532.1.1.11.7 GrCapabilities

[advertised]	*GrCapability* see 2.532.1.1.11.8	Graceful Restart capability advertised to peer.
[ipv4VpnCap]	*Capability* see 2.532.1.1.11.5	Graceful Restart IPv4 VPN capability.
[received]	*GrCapability* see 2.532.1.1.11.8	Graceful Restart capability received from peer.

2.532.1.1.11.8 GrCapability

bgpRestarted	*boolean*	BGP restarted.

[forwardingStatesPreserved]	*ForwardingStates* see 2.532.1.1.11.9	Graceful restart capabilities forwarding states are preserved.
gracefulNotification-Supported	*boolean*	Graceful notification is supported.
restartTime	*floating point number*	Restart Time.

2.532.1.1.11.9 ForwardingStates

[dps]	*boolean*	Dynamic Path Selection forwarding state is preserved.
[ipv4FlowSpec] **	*boolean*	IPv4 flow specification forwarding state is preserved. **Since: 4.28.2
[ipv4FlowSpecVpn] **	*boolean*	VPN-IPv4 flow specification forwarding state is preserved. **Since: 4.30.0
[ipv4MplsLabels]	*boolean*	IPv4 labeled-unicast forwarding state is preserved.
[ipv4MplsVpn]	*boolean*	VPN-IPv4 forwarding state is preserved.
[ipv4Multicast]	*boolean*	IPv4 multicast forwarding state is preserved.
[ipv4Mvpn] **	*boolean*	IPv4 MVPN forwarding state is preserved. **Since: 4.28.2
[ipv4SrTe]	*boolean*	IPv4 SR-TE forwarding state is preserved.
[ipv4Unicast]	*boolean*	IPv4 unicast forwarding state is preserved.
[ipv6FlowSpec] **	*boolean*	IPv6 flow specification forwarding state is preserved. **Since: 4.28.2
[ipv6FlowSpecVpn] **	*boolean*	VPN-IPv6 flow specification forwarding state is preserved. **Since: 4.30.0
[ipv6MplsLabels]	*boolean*	IPv6 labeled-unicast forwarding state is preserved.
[ipv6MplsVpn]	*boolean*	VPN-IPv6 forwarding state is preserved.
[ipv6Multicast]	*boolean*	IPv6 multicast forwarding state is preserved.
[ipv6SrTe]	*boolean*	IPv6 SR-TE forwarding state is preserved.
[ipv6Unicast]	*boolean*	IPv6 unicast forwarding state is preserved.
[l2VpnEvpn]	*boolean*	L2VPN EVPN forwarding state is preserved.
[l2VpnVpls] **	*boolean*	L2VPN VPLS forwarding state is preserved. **Since: 4.28.1
[linkState]	*boolean*	BGP Link State forwarding state is preserved.
[rtMembership]	*boolean*	RT Membership forwarding state is preserved.

2.532.1.1.11.10 LlgrCapabilities

[advertised]	*LlgrCapability* see 2.532.1.1.11.11	Long Lived Graceful Restart advertised capabilities.
[received]	*LlgrCapability* see 2.532.1.1.11.11	Long Lived Graceful Restart received capabilities.

2.532.1.1.11.11 LlgrCapability

[dps]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	Dynamic Path Selection capability.
[helperOnly]	*boolean*	Helper only.
[ipv4FlowSpec]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 flow specification capability.
[ipv4FlowSpecVpn] **	*LlgrCapabilityInfo* see 2.532.1.1.11.12	VPN-IPv4 flow specification capability. **Since: 4.30.0
[ipv4MplsLabels]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 labeled-unicast capability.
[ipv4MplsVpn]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	VPN-IPv4 capability.
[ipv4Multicast]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 multicast capability.
[ipv4Mvpn] **	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 MVPN capability. **Since: 4.28.2
[ipv4SrTe]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 SR-TE capability.
[ipv4Unicast]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv4 unicast capability.
[ipv6FlowSpec]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv6 flow specification capability.
[ipv6FlowSpecVpn] **	*LlgrCapabilityInfo* see 2.532.1.1.11.12	VPN-IPv6 flow specification capability. **Since: 4.30.0
[ipv6MplsLabels]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv6 labeled-unicast capability.
[ipv6MplsVpn]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	VPN-IPv6 capability.
[ipv6Multicast]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv6 multicast capability.
[ipv6SrTe]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv6 SR-TE capability.
[ipv6Unicast]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	IPv6 unicast capability.
[l2VpnEvpn]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	L2VPN EVPN capability.
[l2VpnVpls] **	*LlgrCapabilityInfo* see 2.532.1.1.11.12	L2VPN VPLS capability. **Since: 4.28.1
[linkState]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	BGP Link State capability.
[rtMembership]	*LlgrCapabilityInfo* see 2.532.1.1.11.12	RT Membership capability.

2.532.1.1.11.12 LlgrCapabilityInfo

[forwardingStatePreserved]	*boolean*	Forwarding State is preserved.

staleTime	*integer*	Stale time (Seconds).

2.532.1.1.11.13 MultipleLabelsCapabilities

[ipv4MplsLabels]	*MultipleLabelsCapability* see 2.532.1.1.11.14	IPv4 MPLS Label.
[ipv6MplsLabels]	*MultipleLabelsCapability* see 2.532.1.1.11.14	IPv6 MPLS Label.

2.532.1.1.11.14 MultipleLabelsCapability

[advertised]	*integer*	Advertised count.
[received]	*integer*	Received count.

2.532.1.1.11.15 MultiprotocolCapabilities

[dps]	*Capability* see 2.532.1.1.11.5	Dynamic Path Selection capability.
[ipv4FlowSpec]	*Capability* see 2.532.1.1.11.5	IPv4 flow specification capability.
[ipv4FlowSpecVpn] **	*Capability* see 2.532.1.1.11.5	VPN-IPv4 flow specification capability. **Since: 4.30.0
[ipv4MplsLabels]	*Capability* see 2.532.1.1.11.5	IPv4 labeled-unicast capability.
[ipv4MplsVpn]	*Capability* see 2.532.1.1.11.5	VPN-IPv4 capability.
[ipv4Multicast]	*Capability* see 2.532.1.1.11.5	IPv4 multicast capability.
[ipv4Mvpn] **	*Capability* see 2.532.1.1.11.5	IPv4 MVPN capability. **Since: 4.28.2
[ipv4SrTe]	*Capability* see 2.532.1.1.11.5	IPv4 SR-TE capability.
[ipv4Unicast]	*Capability* see 2.532.1.1.11.5	IPv4 unicast capability.
[ipv6FlowSpec]	*Capability* see 2.532.1.1.11.5	IPv6 flow specification capability.
[ipv6FlowSpecVpn] **	*Capability* see 2.532.1.1.11.5	VPN-IPv6 flow specification capability. **Since: 4.30.0
[ipv6MplsLabels]	*Capability* see 2.532.1.1.11.5	IPv6 labeled-unicast capability.
[ipv6MplsVpn]	*Capability* see 2.532.1.1.11.5	VPN-IPv6 capability.
[ipv6Multicast]	*Capability* see 2.532.1.1.11.5	IPv6 multicast capability.
[ipv6SrTe]	*Capability* see 2.532.1.1.11.5	IPv6 SR-TE capability.
[ipv6Unicast]	*Capability* see 2.532.1.1.11.5	IPv6 unicast capability.
[l2VpnEvpn]	*Capability* see 2.532.1.1.11.5	L2VPN EVPN capability.

[l2VpnVpls] **	*Capability* see 2.532.1.1.11.5	L2VPN VPLS capability. **Since: 4.28.1
[linkState]	*Capability* see 2.532.1.1.11.5	BGP Link State capability.
[rtMembership]	*Capability* see 2.532.1.1.11.5	RT Membership capability.

2.532.1.1.12 PeerDropStats

[inDropAsloop]	*integer*	In packets dropped due to Asloop.
[inDropClusterIdLoop] **	*integer*	In packets dropped due to Cluster ID loop. **Since: 4.29.2
[inDropEnforceFirstAs]	*integer*	In packets dropped due to Enforce First As.
[inDropMalformedMpbgp]	*integer*	In packets dropped due to Malformed Mp-Bgp.
[inDropNhAfV6]	*integer*	In packets dropped due to next hop AF V6.
[inDropNhInvalid]	*integer*	In packets dropped due to next hop invalid.
[inDropNhLocal]	*integer*	In packets dropped due to next hop local.
[inDropOrigId]	*integer*	In packets dropped due to Orig Id.
[outDropV4LocalAddr]	*integer*	Out packets dropped due to V4 Local Address.
[outDropV6LocalAddr]	*integer*	Out packets dropped due to V6 Local Address.
[prefixBgpLsDroppedReceptionUnsupported]	*integer*	Link-state NLRIs dropped due to unsupported route type.
[prefixDroppedMartianV4]	*integer*	IPv4 NLRIs dropped due to Martian prefix.
[prefixDroppedMartianV6]	*integer*	IPv6 NLRIs dropped due to Martian prefix.
[prefixEvpnDroppedImportMatchFailure]	*integer*	L2VPN EVPN NLRIs dropped due to route import match failure.
[prefixEvpnDroppedUnsupportedRouteType]	*integer*	L2VPN EVPN NLRIs dropped due to unsupported route type.
[prefixLuDroppedMartianV4]	*integer*	IPv4 LU NLRIs dropped due to Martian prefix.
[prefixLuDroppedMartianV6]	*integer*	IPv6 LU NLRIs dropped due to Martian prefix.
[prefixLuDroppedV4]	*integer*	Packets dropped due to V4 Prefix LU.
[prefixLuDroppedV6]	*integer*	Packets dropped due to V6 Prefix LU.
[prefixRtMembershipDroppedLocalAsReject]	*integer*	RT Membership NLRIs dropped due to local origin ASN received from external peer.
[prefixVpnIpv4DroppedImportMatchFailure]	*integer*	VPN-IPv4 NLRIs dropped due to route import match failure.
[prefixVpnIpv6DroppedImportMatchFailure]	*integer*	VPN-IPv6 NLRIs dropped due to route import match failure.

2.532.1.1.13 PerAfiSafiInfo

[apSendLimit] **	*integer*	Additional-path send limit. **Since: 4.28.1
[apSendMode] **	*string*	Additional-path send mode. **Since: 4.28.1

[apSendPrefixList] **	*string*	Additional-path send prefix-list. **Since: 4.28.1
[localNexthop]	*string*	Local next hop address for next hop self.
[maxTotalAcceptedRoutes] **	*integer*	Maximum number of accepted routes (0 means unlimited). **Since: 4.28.1
[maxTotalAdvertisedRoutes] **	*integer*	Maximum number of advertised routes (0 means unlimited). **Since: 4.28.1
[maxTotalRoutes] **	*integer*	Maximum number of received routes (0 means unlimited). **Since: 4.28.1
[totalAcceptedRoutesWarnLimit] **	*integer*	Number of accepted routes after which a warning is issued (0 means never warn). **Since: 4.28.1
[totalAdvertisedRoutesWarnLimit] **	*integer*	Number of advertised routes after which a warning is issued (0 means never warn). **Since: 4.28.1
[totalRoutesWarnLimit] **	*integer*	Number of received routes after which a warning is issued (0 means never warn). **Since: 4.28.1

2.532.1.1.14 RouteFilterInfo

filterType	*enum: RouteMap | PrefixList | RCF*	Type of route filter.
[inboundIpv4SrTe]	*string*	Inbound IPv4 SR-TE.
[inboundIpv4Uni]	*string*	Inbound IPv4 Unicast.
[inboundIpv6SrTe]	*string*	Inbound IPv6 SR-TE.
[inboundIpv6Uni]	*string*	Inbound IPv6 Unicast.
[outboundIpv4Uni]	*string*	Outbound IPv4 Unicast.
[outboundIpv6Uni]	*string*	Outbound IPv6 Unicast.

2.532.1.1.15 RouteMapFilterInfo

filterType	*RouteMap*	Type of route filter.
[inboundIpv4SrTe]	*string*	Inbound IPv4 SR-TE.
[inboundIpv4Uni]	*string*	Inbound IPv4 Unicast (overridden if RCF is configured).
[inboundIpv6SrTe]	*string*	Inbound IPv6 SR-TE.
[inboundIpv6Uni]	*string*	Inbound IPv6 Unicast (overridden if RCF is configured).
[outboundIpv4Uni]	*string*	Outbound IPv4 Unicast (overridden if RCF is configured).
[outboundIpv6Uni]	*string*	Outbound IPv6 Unicast (overridden if RCF is configured).

2.532.1.1.16 TcpAoInfo

[activeKeyIds]	*list of integer*	All active key IDs.
[currentKeyId]	*integer*	Key ID of the current TCP-AO key.

macAlgorithm	*enum: aes128Cmac96 | hmacSha196*	MAC computation algorithm.
[receiveNextKeyId]	*integer*	Key ID of the receive next TCP-AO key.
sharedSecretProfile	*string*	TCP-AO shared secret profile.
sharedSecretProfileState **	*enum: valid | noValidSecrets | notExist*	State of the Shared Secret Profile. **Since: 4.28.2

2.533 show ip bgp neighbors update errors [...]

revision 2

vrfs	*table(string) = BgpNeighborUpdateError* see 2.533.1	Per VRF BGP Neighbor Update-Error list.

2.533.1 BgpNeighborUpdateError

peers	*table(string) = BgpUpdateErrorPeerEntry* see 2.533.1.1	Dictionary of BGP Update Error entries indexed by the peer address.
vrf	*string*	VRF Name.

2.533.1.1 BgpUpdateErrorPeerEntry

[disableAfiSafiAttr]	*string*	Disable AFI/SAFI Attribute.
[disableAfiSafiBuffer]	*string*	Disable AFI SAFI Update Buffer.
[ignoreAttr]	*string*	Ignore Attribute.
[ignoreBuffer]	*string*	Ignore Attribute Update Buffer.
[withdrawAttr]	*string*	Withdraw Attribute.
[withdrawBuffer]	*string*	Withdraw Update Buffer.

2.534 show ip bgp paths [...]

vrfs	*table(string) = BgpASPathList* see 2.534.1	Per VRF BGP AS path info list.

2.534.1 BgpASPathList

asPathList	*list of BgpASPathEntry* see 2.534.1.1	List of AS path entries.

2.534.1.1 BgpASPathEntry

asPathInfo	*string*	AS path string.
med	*integer*	Multi Exit Discriminator for the route.
referenceCount	*integer*	Number of references to the AS path.

2.535 show ip bgp regex REGEX [...]

revision 3

see 2.17

2.536 show ip bgp summary [...]

revision 3

see 2.46

2.537 show ip community-list [...]

ipCommunityLists	*table(string) = IpCommunityList* see 2.537.1	Table of community lists by name.

2.537.1 IpCommunityList

entries	*list of IpCommunityListEntry* see 2.537.1.1	Community list entries sorted by increasing sequence number.

2.537.1.1 IpCommunityListEntry

[communityRegex]	*string*	Communities as a regular expression.
[communityValues]	*list of string*	Community list values.
filterType	*enum: permit | deny*	Filter type.
listType	*enum: expanded | standard*	Community list type.

2.538 show ip decap-group [...]

revision 4

decapGroups	*table(string) = DecapGroupModelBase* see 2.538.1	A mapping from name to ip decap-group entry.
globalUdpDestPortToPayload-Type	*table(numeric string) = string*	A global mapping from UDP destination port to payload type.

2.538.1 DecapGroupModelBase

Could also be an instance of:
DecapGroupGreModel 2.538.2
DecapGroupIpIpModel 2.538.3
DecapGroupQosTcModel 2.538.4
DecapGroupUdpModel 2.538.5

[forwardingVrf]	*string*	Forwarding VRF.
persistent	*boolean*	Decap group is present in system configuration.
tunnelType	*enum: ipip | gre | udp | unknown*	Decap tunnel type.

2.538.2 DecapGroupGreModel

decapIntf	*list of DecapIntfModel* see 2.538.2.2	decap intf key list.
decapIp	*list of IP address prefix*	decap ip list.
[decapIpCounters]	*table(IPv4 or IPv6 address prefix) = DecapGroupCounterEntry* see 2.538.2.1	Map decap IP address to decap group counter.
[forwardingVrf]	*string*	Forwarding VRF.
persistent	*boolean*	Decap group is present in system configuration.
tunnelType	*enum: ipip | gre | udp | unknown*	Decap tunnel type.

2.538.2.1 DecapGroupCounterEntry

octets	*integer*	Number of received octets on Decap-Group.
packets	*integer*	Number of received packets on Decap-Group.

2.538.2.2 DecapIntfModel

addressFamily	*enum: ipv4 | ipv6 | ipunknown*	Decap address family.
addressType	*enum: all | primary*	Decap address type.
[counter]	*DecapGroupCounterEntry* see 2.538.2.1	Counter for the decap interface.
intfId	*interface name*	Decap destination interface.

oldConfig	*boolean*	Old way of writing this config.

2.538.3 DecapGroupIpIpModel

decapIntf	*list of DecapIntfModel* see 2.538.2.2	decap intf key list.
decapIp	*list of IP address prefix*	decap ip list.
[decapIpCounters]	*table(IPv4 or IPv6 address prefix) = DecapGroupCounterEntry* see 2.538.2.1	Map decap IP address to decap group counter.
[forwardingVrf]	*string*	Forwarding VRF.
persistent	*boolean*	Decap group is present in system configuration.
tunnelType	*enum: ipip | gre | udp | unknown*	Decap tunnel type.

2.538.4 DecapGroupQosTcModel

[forwardingVrf]	*string*	Forwarding VRF.
persistent	*boolean*	Decap group is present in system configuration.
qosTcFromMplsTc	*boolean*	QoS Traffic Class derived from MPLS Traffic Class.
tunnelType	*enum: ipip | gre | udp | unknown*	Decap tunnel type.

2.538.5 DecapGroupUdpModel

decapIntf	*list of DecapIntfModel* see 2.538.2.2	decap intf key list.
decapIp	*list of IP address prefix*	decap ip list.
destinationPort	*integer*	Decap destination port.
[forwardingVrf]	*string*	Forwarding VRF.
payloadType	*enum: unknownPayload | ipv6 | ipv4 | ipvx | mpls*	Decap Payload Type.
persistent	*boolean*	Decap group is present in system configuration.
tunnelType	*enum: ipip | gre | udp | unknown*	Decap tunnel type.

2.539 show ip decap-group qos [...]

groups	*table(string) = DecapGroupQosTcModel* see 2.538.4	IP decap groups, keyed by their name.

2.540 show ip dhcp relay

revision 2

see 2.55

2.541 show ip dhcp snooping

[bridgeMac]	*MAC address*	Remote MAC address.
bridgingEnabled **	*boolean*	DHCP Snooping with bridging enabled. **Since: 4.25.2
[circuitIdEnabled]	*boolean*	Circuit ID is enabled.
[circuitIdFormat]	*enum: %h:%p | %p:%v*	Format for circuit-id.
[circuitIdType]	*integer*	Circuit-id default sub-option type.
[circuitIds]	*table(Interface name) = CircuitIdModel* see 2.541.1	Circuit IDs.
enabled	*boolean*	DHCP Snooping is enabled.
enabledVlans	*list of integer*	VLANs on which snooping is configured.
operational	*boolean*	DHCP Snooping is operational.
operationalVlans	*list of integer*	VLANs on which snooping is operational.
option82Enabled	*boolean*	Insertion of Option-82 is enabled.

2.541.1 CircuitIdModel

[circuitType]	*integer*	Circuit ID type.
name	*interface name*	Interface name.
valid	*boolean*	Circuit ID is valid.
value	*string*	Circuit ID value.

2.542 show ip dhcp snooping counters [...]

detail_ **	*boolean*	Counter values in detail. **Since: 4.25.2
enabled	*boolean*	DHCP Snooping is enabled.
vlanCounters	*table(numeric string) = VlanCounters* see 2.542.1	Per-VLAN counter information.

2.542.1 VlanCounters

interfaces **	*table(Interface name) = IntfCounters* see 2.542.1.1	Vlan specific L2 interface counters. **Since: 4.25.2
repliesDropped	*integer*	Replies dropped.
repliesForwarded	*integer*	Replies forwarded.
repliesReceived	*integer*	Replies received.
requestsDropped	*integer*	Requests dropped.
requestsForwarded	*integer*	Requests forwarded.
requestsReceived	*integer*	Requests received.
resetTime	*floating point number*	Last time counters were reset.
vlan	*integer*	VLAN ID.

2.542.1.1 IntfCounters

repliesDropped	*integer*	Replies dropped.
repliesForwarded	*integer*	Replies forwarded.
repliesReceived	*integer*	Replies received.
requestsDropped	*integer*	Requests dropped.
requestsForwarded	*integer*	Requests forwarded.
requestsReceived	*integer*	Requests received.

2.543 show ip dhcp snooping counters debug

[dhcpOpErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Dhcp Optype.
[disabledErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Snooping disabled.
enabled	*boolean*	DHCP Snooping is enabled.
[forwardedCounters]	*SnoopingRelayCounters* see 2.543.1	Forwarded.
[infoOptErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Info Option.
[parseErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Parse error.
[receivedCounters]	*SnoopingRelayCounters* see 2.543.1	Recieved.
[remoteIdErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Remote-ID Option.
[resetTime]	*floating point number*	Last time counters were reset.

[vlanIdErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid VlanId.

2.543.1 SnoopingRelayCounters

relayToSnooping	*integer*	Relay to snooping counter.
snoopingToRelay	*integer*	Snooping to relay counter.

2.544 show ip dhcp snooping hardware

enabled	*boolean*	DHCP Snooping is enabled.
vlans	*list of integer*	VLANs on which snooping is enabled.
vlansPerSlice	*table(string) = VlanList* see 2.544.1	VLANs enabled per slice.

2.544.1 VlanList

vlans	*list of integer*	VLANs in slice.

2.545 show ip domain-name

see 2.301

2.546 show ip extcommunity-list [...]

ipExtCommunityLists	*table(string) = IpExtCommunityList* see 2.546.1	Table of extended community lists by name.

2.546.1 IpExtCommunityList

entries	*list of IpCommunityListEntry* see 2.537.1.1	Extended community list entries sorted by increasing sequence number.

2.547 show ip hardware ale adj [...]

[adj]	*table(numeric string) = AleL3AdjInfo* see 2.547.1	Dictionary to map fecId to L3Adj.
[cmdErr]	*string*	Error Message.
[transientAdjsPresent] **	*boolean*	Output contains some transient adjs. **Since: 4.18.1

2.547.1 AleL3AdjInfo

[adjRefCount] **	*integer*	Inter Adjacency references. **Since: 4.22.0
[adjStats]	*AleStats* see 2.547.1.2	SubModel storing Adjacency Timestamps.
[backupL3AdjVia] **	*list of AleL3AdjVia* see 2.50.1.1	List storing the backup L3 V4 Adjacency Vias. **Since: 4.23.2
[backupL3AdjVia6] **	*list of AleL3AdjVia* see 2.50.1.1	List storing the backup L3 V6 Adjacency Vias. **Since: 4.23.2
[extRefCount] **	*integer*	External FEC references. **Since: 4.22.0
[fecRefs] **	*list of AlePrevLevelFecs* see 2.547.1.1	List of previous level fec refs. **Since: 4.24.2
[l2AdjIndex]	*integer*	Layer 2 Adjacency Index.
[l3AdjVia]	*list of AleL3AdjVia* see 2.50.1.1	List storing the L3 V4 Adjacency Vias.
[l3AdjVia6]	*list of AleL3AdjVia* see 2.50.1.1	List storing the L3 V6 Adjacency Vias.
[orderedNexthops] **	*boolean*	The ordering of next-hops is honored. **Since: 4.24.1
[pbrRefCount] **	*integer*	PBR references. **Since: 4.22.0
[prefixCount]	*integer*	Number of Prefixes.
[proxyRefCount] **	*integer*	proxy FEC references. **Since: 4.24.2

[routeType]	*enum: usedByTunnelRoute | attachedRoute | directedBcastRoute | tunnelFibRoute | dropRoute | receiveRoute | routeCacheForwardRoute | evpnRoute | routeCacheConnectedRoute | routeTypeUnknown | invalidRoute | policyRoute | fecInfoUnavailable | nexthopRoute | mplsRoute | mplsFloodSetRoute | bcastReceiveRoute | l2NexthopGroupRoute | nexthopGroupRoute | forwardRoute | vplsMcastRoute | slowReceiveRoute | pwRoute | vplsL2UcastRoute | mplsMcastRoute | greMcastRoute*	Specifies Route Type.
[sourceFec] **	*integer*	Source FEC being used by this proxy. **Since: 4.24.2
[totalRefCount] **	*integer*	Total references. **Since: 4.22.0
[transientAdj] **	*boolean*	ECMP Adj programmed as non ECMP in hardware. **Since: 4.18.1
[v4PrefixesByVrf] **	*table(string) = AleV4Prefixes* see 2.547.1.3	Dict to list V4 Prefix references by VRF. **Since: 4.22.0
[v6PrefixesByVrf] **	*table(string) = AleV6Prefixes* see 2.547.1.4	Dict to list V6 Prefix references by VRF. **Since: 4.22.0

2.547.1.1 AlePrevLevelFecs

fecId	*integer*	Previous level FEC ID.
tableId	*integer*	FEC table ID.

2.547.1.2 AleStats

insertTime	*floating point number*	Insert timestamp.
lastUpdateTime	*floating point number*	Update timestamp.
numUpdates	*integer*	Number of updates.

2.547.1.3 AleV4Prefixes

[v4Prefixes]	*list of AlePrefix* see 2.547.1.3.1	A mapping between VRF ID and V4 Prefixes.

2.547.1.3.1 AlePrefix

prefix	*string*	Prefix.

2.547.1.4 AleV6Prefixes

[v6Prefixes]	*list of AlePrefix* see 2.547.1.3.1	A mapping between VRF ID and V6 Prefixes.

2.548 show ip hardware ale adj acknowledgement [...]

[cmdErr]	*string*	Error Message.
fecEntries	*table(numeric string) = AcknowledgementFecEntry* see 2.548.1	Mapping fecId to its version ID info.
vrfName	*string*	VRF name.

2.548.1 AcknowledgementFecEntry

[acknowledged]	*AcknowledgementFecVersionId* see 2.548.1.1	FEC version Id that was acknowledged.
[requested]	*AcknowledgementFecVersionId* see 2.548.1.1	FEC version Id to be acknowledged.

2.548.1.1 AcknowledgementFecVersionId

failed **	*boolean*	FEC install failed. **Since: 4.28.0
requestedDelete	*boolean*	FEC is due to be deleted.
versionId	*integer*	FEC version Id to be acknowledged.

2.549 show ip hardware ale adj export [...]

[cmdErr]	*string*	Error message.
[fecs]	*table(numeric string) = AleOpenConfigExportedFec* see 2.549.1	A mapping of FEC IDs to exported FEC information.

2.549.1 AleOpenConfigExportedFec

nextHops	*table(numeric string) = AleOpenConfigExportedNextHop* see 2.549.1.1	A mapping of next-hops IDs to exported next-hop information.
programmedId	*integer*	Exported programmed ID.

2.549.1.1 AleOpenConfigExportedNextHop

interface	*interface name*	Interface.
ipAddress	*IP address*	IP address.
[labels]	*list of integer*	MPLS labels.
macAddress	*MAC address*	MAC address.
popTopLabel	*boolean*	Pop the top label from the MPLS label stack.
weight	*integer*	Next hop weight.

2.550 show ip hardware ale adj proxy [...]

[cmdErr]	*string*	Error Message.
[fecAliases]	*list of FecProxyInfo* see 2.550.1	List of FEC proxies.

2.550.1 FecProxyInfo

[cmdErr]	*string*	Error Message.
proxyFecList	*list of integer*	List of proxy FECs for source.
sourceFec	*integer*	Source Fec Id for this proxy set.

2.551 show ip hardware ale counters egress bindings l2-adj [...]

[cmdErr]	*string*	Error Message.
[l2AdjCounterBindings]	*table(numeric string) = AleL2AdjCounterBindingsInfo* see 2.551.1	Dictionary to map Layer 2 adjacency index to its counter related bindings.

2.551.1 AleL2AdjCounterBindingsInfo

pdcks	*list of AlePdckValues* see 2.551.1.1	List of platform-dependent counter keys bound to the L2 Adjacency.

2.551.1.1 AlePdckValues

value1	*integer*	PDCK value 1.
value2	*integer*	PDCK value 2.

2.552 show ip hardware ale counters egress bindings platform-key [...]

[cmdErr]	*string*	Error Message.
[pdcksCounterBindings]	*table(string) = AlePdckCounterBindingsInfo* see 2.552.1	Dictionary to map Ale Platform-dependent counter key (PDCK) to PDCK counter related bindings.

2.552.1 AlePdckCounterBindingsInfo

pick	*integer*	Platform-independent counter key bound to the PDCK.
pickIndex	*integer*	PICK Index.
[pickType]	*enum: counterKeyTypeMax* | *tunnelCounter* | *nhgCounterUnshared* | *prefixCounterPrimary* | *nhgCounterShared* | *prefixCounterBackup* | *invalidCounterKeyType*	Specifies counter key Type.

2.553 show ip hardware ale l2-adj [...]

[cmdErr]	*string*	Error Message.
[l2Adj]	*table(numeric string) = AleL2AdjInfo* see 2.553.1	Dictionary to map l2AdjIndex to l2Adj Info.

2.553.1 AleL2AdjInfo

[adjCount]	*integer*	Number of L3 Adjacencies.
[backupL2AdjVia] **	*list of AleL2AdjVia* see 2.553.1.1	List storing the backup L2 Adjacency Vias. **Since: 4.23.2
l2AdjState **	*enum: ecmpGroupsFull* | *installed* | *pendingSync* | *pendingDelete* | *installNotReady* | *installFailed* | *garbageCollected* | *installNotRequired* | *uninitialized*	L2 Adjacency State. **Since: 4.21.0
[l2AdjStats] **	*AleStats* see 2.547.1.2	L2 Adjacency timestamps. **Since: 4.21.3
[l2AdjVia]	*list of AleL2AdjVia* see 2.553.1.1	List storing the L2 Adjacency Vias.
[platformFecIndex] **	*integer*	Platform FEC table index used by the L2 Adjacency. **Since: 4.21.0

2.553.1.1 AleL2AdjVia

adjType	*enum: nextHopCpuRouteCache* | *nextHopPopLabelAndRoute* | *nextHopFlow* | *nextHopVlanPort* | *nextHopTunnel* | *nextHopArpDrop* | *nextHopDrop* | *nextHopHierarchical* | *nextHopDsfTunnel* | *nextHopCpuSlow* | *nextHopCpuBcast* | *nextHopMcastTunnel* | *nextHopCpu* | *nextHopDecapFallback* | *nextHopRouteOverflow-Drop* | *nextHopArpRequired* | *nextHopArpResolved*	L2 Adjacency Type.
[counterId] **	*integer*	Next-Hop Counter Id. **Since: 4.23.0
[fallbackInfo] **	*FallbackInfo* see 2.50.1.1.2	Fallback information. **Since: 4.27.2
[greEncapInfo] **	*GreEncapInfo* see 2.553.1.1.2	GRE Encap information. **Since: 4.24.2

[ipInIpEncapInfo] **	*IpInIpEncapInfo* see 2.553.1.1.3	IP-in-IP Encap information. **Since: 4.28.2
[ipTunnelInfo] **	*IpTunnelInfo* see 2.553.1.1.1	Tunnel resolution information. **Since: **currentRelease**
l2IntfId	*interface name*	L2 Interface Id.
l3IntfId	*interface name*	L3 Interface Id.
macAddr	*MAC address*	Next-Hop Mac Address.
[mplsEncapInfo] **	*MplsEncapInfo* see 2.553.1.1.4	MPLS Encap information. **Since: 4.24.2
[nextLevelL2AdjIndex] **	*integer*	L2 Adjacency ID of the next FEC in a hierarchical FEC. **Since: 4.21.0
[vxlanEncapInfo] **	*VxlanEncapInfo* see 2.553.1.1.5	VXLAN Encap information. **Since: 4.26.2
weight	*integer*	Metric representing the hop cost.

2.553.1.1.1 IpTunnelInfo

destination	*IP address*	Destination IP address.
[interface]	*interface name*	Tunnel interface.
source	*IP address*	Source IP address.
[underlayL2Adj]	*integer*	L2 Adjacency ID of the underlay FEC.

2.553.1.1.2 GreEncapInfo

destination	*IP address prefix*	Location in the IP network where the tunnel terminates.
source	*IP address prefix*	Location in the IP network where the tunnel starts.
[tos]	*integer*	Type of service.
[ttl]	*integer*	Time to live.

2.553.1.1.3 IpInIpEncapInfo

destination	*IP address prefix*	Destination IP address.
source	*IP address prefix*	Source IP address.

2.553.1.1.4 MplsEncapInfo

[entropyLabelIndicatorIndices] **	*list of integer*	Entropy label indicator indices in the label stack of the tunnel. **Since: 4.28.1
[entropyLabelIndices] **	*list of integer*	Entropy label indices in the label stack of the tunnel. **Since: 4.28.1
labelStack	*list of integer*	MPLS Label Stack.
operation	*enum: swap | push | pop | forward*	Label Operation.

2.553.1.1.5 VxlanEncapInfo

srcVti	*interface name*	Source VTI.
vtepIp	*IP address*	VTEP IP address.

2.554 show ip hardware ale mac-address router gateway

macs	*table(string) = VlanListType* see 2.554.1	Mapping MAC address to its VLANs.

2.554.1 VlanListType

vlans	*list of integer*	VLAN IDs.

2.555 show ip hardware ale routes [...]

[cmdErr]	*string*	Error Message.
[fibRoute]	*table(IPv4 or IPv6 address prefix) = AleRouteInfo* see 2.555.1	Mapping prefix to its FibRoute Info.
[numUnprog]	*integer*	Count of unprogrammed routes.
[vrfName]	*string*	VRF name.

2.555.1 AleRouteInfo

[cmdErr]	*string*	Error Message.
[l2AdjInfo]	*table(numeric string) = AleL2AdjInfo* see 2.553.1	Dict to handle L2 Adjacency Info.
[l3AdjInfo]	*table(numeric string) = AleL3AdjInfo* see 2.547.1	Dict to handle L3 Adjacency Info.
[parent]	*string*	Parent prefix.
[routeOptimized]	*enum: optimizedAndCompressed | optimizedAndExpanded | optimized | unoptimized*	Details of Route Optimization.
[routeProgrammed]	*boolean*	Flag to indiacte Route is programmed or not.
[routeStats]	*AleStats* see 2.547.1.2	Stats of a route.

2.556 show ip hardware ale routes vrf all [...]

[cmdErr]	*string*	Error Message.
[numUnprog]	*integer*	Count of unprogrammed routes in all VRFs.

[vrfs]	*table(string) = AleRoute* see 2.555	Dictionary mapping VRF names to ale routes in that VRF.

2.557 show ip hardware ale vrf

vrfStatus	*table(string) = VrfStatus* see 2.557.1	List of all Vrf names and IDs.

2.557.1 VrfStatus

tableId **	*integer*	ID used in Ale for the FIB table. **Since: 4.21.3
vrfId	*integer*	ID used on hardware for the VRF.
vrfName	*string*	Name of a VRF.

2.558 show ip hardware fib ecmp resilience [...]

[policyBasedResRouteInfos] **	*AleResEcmpRouteInfos* see 2.558.1	Policy marked resilient routes information. **Since: 4.29.1
resPrefixesToRouteInfos	*table(IPv4 or IPv6 address prefix) = AleResEcmpRouteInfos* see 2.558.1	Map from resilient ECMP prefixes to their covered routes.
vrfName	*string*	VRF name.

2.558.1 AleResEcmpRouteInfos

[capacity]	*integer*	The resilient capacity.
[orderedVias] **	*boolean*	The resilient prefix is configured as ordered. **Since: 4.28.1
[redundancy]	*integer*	The resilient redundancy.
routes	*list of AleResEcmpRouteInfo* see 2.558.1.1	List of resilient routes information.

2.558.1.1 AleResEcmpRouteInfo

fecId	*integer*	The non-resilient FEC ID.
orderedVias **	*boolean*	The vias in the route's ECMP FEC are programmed in a deterministic order. **Since: 4.28.1
prefix	*IP address prefix*	The route's prefix.
reason	*string*	Resason why route is not programmed as resilient.
resFecId	*integer*	The resilient FEC ID.
resilient	*boolean*	The route is programmed as resilient.

2.559 show ip hardware fib ecmp resilience summary [...]

numOrdered **	*integer*	Number of ordered resilient routes. **Since: 4.28.1
numPrefixes	*integer*	Number of programmed resilient ECMP prefixes.
numProgrammed	*integer*	Number of routes programmed as resilient.
numUnprogrammed	*integer*	Number of routes not programmed as resilient.
vrfName	*string*	VRF Name.

2.560 show ip hardware fib load-balance distribution

dlbEcmpNhg	*list of string*	Dynamic load balancing enabled static nexthop groups.
dlbFlowSetSize **	*integer*	DLB flow set size. **Since: 4.28.1
dlbInactivityDuration **	*integer*	DLB inactivity duration in microseconds. **Since: 4.26.2
dlbPortLoadingWeight **	*integer*	DLB average traffic weight. **Since: 4.27.0
dlbSamplingPeriod **	*integer*	DLB sampling period in microseconds. **Since: 4.26.2
globalDlbEcmpEnabled	*boolean*	Whether or not dynamic load balancing is enabled globally.
lbnBasedDlbHashing **	*boolean*	Whether or not load-balance number is used in DLB hashing. **Since: **currentRelease**
lbnBasedStaticEcmpHashing **	*boolean*	Whether or not load-balance number is used in Static ECMP hashing. **Since: **currentRelease**
randomSelectionControlSeed **	*integer*	DLB Hash Seed. **Since: 4.27.2

2.561 show ip hardware fib load-balance number [...]

interfaces	*table(Interface name) = integer*	Mapping between an interface and LBN.

2.562 show ip hardware fib routes [...]

[cmdErr]	*string*	Error flag.
[route]	*table(IPv4 or IPv6 address prefix) = FibFec* see 2.50	Mapping prefix to its FibRoute Info.
[vrfName]	*string*	VRF Name.

2.563 show ip igmp access-list [...]

see 2.53

2.564 show ip igmp groups GROUPADDR [...]

groupList	*list of IgmpGroup* see 2.564.1	List of IGMP groups joined on different interfaces.

2.564.1 IgmpGroup

[expires]	*floating point number*	The timestamp in UTC after which this group expires if there is no new activity.
filterMode	*enum: filterModeExclude | filterModeInclude*	Group Mode for IGMP.
groupAddress	*IPv4 address*	Group Address of interest.
[groupDetail]	*IgmpGroupDetail* see 2.564.1.1	Optional details about this IgmpGroup.
[interfaceAddress] **	*IPv4 address*	Interface Address. **Since: 4.22.0
interfaceName	*interface name*	Interface on which this group is joined.
lastReporter	*IPv4 address*	The IP Address of the latest host who joined this group.
uptime	*floating point number*	The timestamp in UTC at which this group last became active.

2.564.1.1 IgmpGroupDetail

interfaceAddress	*IPv4 address*	Interface IP Address.
sources	*table(IPv4 address) = IgmpGroupSource* see 2.564.1.1.1	Map of Source Addresses for this group.

2.564.1.1.1 IgmpGroupSource

[expires]	*floating point number*	The timestamp in UTC after which this source expires if there is no new activity.
uptime	*floating point number*	The timestamp in UTC at which this source last became active.

2.565 show ip igmp groups [...]

see 2.564

2.566 show ip igmp groups count [...]

count	*integer*	Number of Igmp Groups.
[interfaceName]	*interface name*	Interface on which to count groups.

2.567 show ip igmp groups interface INTF [...]

see 2.564

2.568 show ip igmp groups interface INTF count [...]

see 2.566

2.569 show ip igmp host-proxy (debug | interface) [...]

IgmpHostProxyInterfaces	*table(Interface name) = IgmpHostProxyInterface* see 2.569.1	Map of IGMP host-proxy interfaces.

2.569.1 IgmpHostProxyInterface

[details]	*IgmpHostProxyInterface-Details* see 2.569.1.3	Detailed interface information.
[deviceName]	*string*	Device name.
groupSource	*table(IPv4 address) = GroupDetail* see 2.569.1.1	Mapping between group address and sources in include/exclude mode.
igmpHostProxyAclInfo	*table(string) = IgmpHostProxyAclInfo* see 2.569.1.2	Mapping between ACL name and information about ACL rules.
igmpHostProxyVersion	*integer*	IGMP host-proxy version on this interface.
unsolicitedReportIntvl	*floating point number*	Unsolicited-report interval on this interface in seconds.

2.569.1.1 GroupDetail

excludeSourceList	*list of IPv4 address*	List of exclude sources.
includeSourceList	*list of IPv4 address*	List of include sources.

2.569.1.2 IgmpHostProxyAclInfo

aclInfo	*table(numeric string) = AclInfo* see 2.569.1.2.1	Mapping between sequence number and information about the rule.

2.569.1.2.1 AclInfo

[action]	*string*	Action on the ACL rule(permit/deny).
[errorMsg]	*string*	Message describing the error in the rule, if any.
[group]	*string*	IpAddress of the group in the rule.
[source]	*string*	IpAddress of the source in the rule.

2.569.1.3 IgmpHostProxyInterfaceDetails

v1QueryReceived	*integer*	IGMP V1 Queries recieved.
v1ReportsSent	*integer*	IGMP V1 reports sent.
v2GeneralQueryReceived	*integer*	IGMP V2 General Queries received.
v2GroupQueryReceived	*integer*	IGMP V2 Group Queries received.
v2ReportsSent	*integer*	IGMP V2 reports sent.
v3GeneralQueryReceived	*integer*	IGMP V3 General Queries received.
v3GroupQueryReceived	*integer*	IGMP V3 Group Queries received.
v3GroupSourceQueryReceived	*integer*	IGMP V3 Group-Source specific received.

v3ReportsSent	*integer*	IGMP V3 reports sent.

2.570 show ip igmp host-proxy config-sanity

intfMisconfigs	*table(Interface name)* = *IHPPotentialMisconfigs* see 2.570.1	Mapping between interface and the IHP potential misconfigs on it.

2.570.1 IHPPotentialMisconfigs

[denyIpMrouteAcls]	*list of string*	Access-lists with "deny ip any any" rule.
[overlappingGroups]	*list of IpAddrAndMask* see 2.241.1.2.2.9	Groups with overlapping permit and deny configurations.
[sourceFilteredGroups]	*list of IpAddrAndMask* see 2.241.1.2.2.9	Groups with include or exclude sources.
[version]	*enum: 1 | 2*	IGMP Host-Proxy configured as IGMP version (1 or 2).

2.571 show ip igmp interface [...]

IgmpInterfaces	*table(Interface name) = IgmpInterface* see 2.571.1	Map of IGMP enabled interfaces.

2.571.1 IgmpInterface

[generalQueryExpiryTime]	*floating point number*	IGMP general query timer expiry in seconds.
igmpEnabled	*boolean*	Denotes whether IGMP is enabled on this interface.
igmpQuerier	*IPv4 address*	Ip Address of the IGMP Querier on this interface.
igmpQueryInterval	*integer*	IGMP Query interval in seconds.
interfaceAddress	*IPv4 address*	Ip Address of this interface.
lastMemberQueryResponse-Count	*integer*	IGMP last member query response count.
lastMemberQueryResponse-Interval	*floating point number*	IGMP last member query response interval in seconds.
maxQueryResponse	*floating point number*	IGMP max query response interval in seconds.
multicastGroups	*list of IPv4 address*	The list of multicast groups joined on this interface.
multicastRoutingEnabled	*boolean*	Denotes whether multicast routing is enabled.
multicastTTL	*integer*	Multicast TTL Threshold.
[otherQuerierExpiryTime]	*floating point number*	Other querier present timer expiry in seconds.
robustness	*integer*	IGMP Robustness on this interface.
routerAlert	*enum: routerAlertMandatory | routerAlertOptional-Connected | routerAlertOptional*	IGMP Router Alert requirement on this interface.
routerVersion	*enum: 1 | 2 | 3 |*	IGMP router version on this interface.
startupQueryCount	*integer*	IGMP startup query count.
startupQueryInterval	*floating point number*	IGMP startup query interval in seconds.

2.572 show ip igmp membership [...]

members	*table(Interface name) = IgmpMemGrpList* see 2.572.1	Map interface to a list of IGMP membership status.

2.572.1 IgmpMemGrpList

groupList	*list of IgmpMemGrp* see 2.572.1.1	List of IGMP membership groups.

2.572.1.1 IgmpMemGrp

excludeSrcs	*list of IP address*	List of Exclude Source Addresses for this group.
groupAddress	*IP address*	Group Address of interest.
includeSrcs	*list of IP address*	List of Include Source Addresses for this group.
interfaceName	*interface name*	Interface on which this group is joined.

2.573 show ip igmp profile [...]

profiles	*table(string) = ProfileInfo* see 2.573.1	Map IGMP Profile to its filter and list of addresses.

2.573.1 ProfileInfo

action	*enum: permit | deny*	Filter type of profile permit/deny.
ranges	*list of Ranges* see 2.573.1.1	Multicast groups covered by the profile.

2.573.1.1 Ranges

max	*IP address*	Maximum address range.
min	*IP address*	Minimum address range.

2.574 show ip igmp snooping [...]

[igmpSnoopingState]	*enum: disabled* | *enabled*	Global state of IGMP Snooping.
[immediateLeave]	*enum: disabled* | *default* | *enabled*	Fast-leave processing global.
[mlagConnectionState] **	*enum: disconnected* | *connected* | *unknown* | *failed* | *connecting* | *disconnecting* | *initializing* | *n/a*	IGMP Snooping agent MLAG connection state. **Since: 4.30.1
[mlagMountState] **	*enum: unknown* | *mounting* | *n/a* | *mounted* | *unmounted* | *failed*	IGMP Snooping agent MLAG mount state. **Since: 4.30.1
[reportFlooding]	*enum: disabled* | *default* | *enabled*	Forwarding of membership report messages.
reportFloodingSwitchPorts	*list of Interface name*	Interfaces for which report-flooding enabled.
robustness	*integer*	Number of queries sent to age out a port's membership in group.
vlans	*table(numeric string) = IgmpSnoopingVlanInfo* see 2.574.1	Map vlan Id to vlan information.

2.574.1 IgmpSnoopingVlanInfo

[evpnProxyActive] **	*boolean*	IGMP proxying enabled via EVPN. **Since: 4.29.2
floodingTraffic	*boolean*	Flooding traffic to VLAN.
groupsOverrun	*boolean*	Indicate if there is an attempt to create group more than maxGroups.
igmpSnoopingState	*enum: disabled* | *enabled*	State of IGMP Snooping.
immediateLeave	*enum: disabled* | *default* | *enabled*	Fast-leave processing on the vlan.
maxGroups	*integer*	Maximum number of multicast groups that can join the vlan.
multicastRouterLearningMode	*pim-dvmrp*	Multicast Router Learning mode.
proxyActive	*boolean*	IGMP snooping proxy active.
pruningActive	*boolean*	True if IGMP snooping pruning active.
reportFlooding	*enum: disabled* | *default* | *enabled*	Forwarding of membership report messages.

2.575 show ip igmp snooping counters [...]

interfaces	*table(Interface name) = IgmpSnoopingCounters-Interface* see 2.575.1	Map Interface name with its counter details.

2.575.1 IgmpSnoopingCountersInterface

badChecksumIgmpPackets-Received	*integer*	Number of packets received for which IGMP checksum check failed.
badChecksumIpPackets-Received	*integer*	Number of packets received for which IP checksum check failed.
badChecksumPimPackets-Received	*integer*	Number of packets received for which PIM checksum check failed.
[dvmrpPacketsReceived]	*integer*	Number of DVMRP packets received.
[igmpLeaveSent]	*integer*	Number of IGMP v2 leaves sent.
[igmpQueriesSent]	*integer*	Number of IGMP queries sent.
[igmpReportSent]	*integer*	Number of IGMP reports sent.
[igmpV1QueryReceived]	*integer*	Number of IGMP v1 queries received.
[igmpV1ReportReceived]	*integer*	Number of IGMP v1 reports received.
[igmpV2LeaveReceived]	*integer*	Number of IGMP v2 leaves received.
[igmpV2QueryReceived]	*integer*	Number of IGMP v2 queries received.
[igmpV2ReportReceived]	*integer*	Number of IGMP v2 reports received.
[igmpV3QueryReceived]	*integer*	Number of IGMP v3 queries received.
[igmpV3ReportReceived]	*integer*	Number of IGMP v3 reports received.
[mrdPacketsReceived]	*integer*	Number of MRD packets received.
nonIpPacketsReceived	*integer*	Number of non IP packets received.
[otherIgmpPacketsReceived]	*integer*	Number of other IGMP packets.
[otherPacketsSent]	*integer*	Number of other packets sent.
[pimPacketsReceived]	*integer*	Number of PIM packets received.
shortPacketsReceived	*integer*	Number of packets received with not enough IP payload.
unknownIpPacketsReceived	*integer*	Number of packets received with unknown IP Protocol.

2.576 show ip igmp snooping groups [...]

member	*table(numeric string) = VlanMembershipInfo* see 2.576.1	Map vlanId to per VLAN group membership information.

2.576.1 VlanMembershipInfo

[groupMembership]	*table(IPv4 or IPv6 address) = GroupMembershipInfo* see 2.576.1.1	Map IGMP group to source membership information.
[routerIntf]	*table(Interface name) = MemberInfo* see 2.576.1.1.1	List of router interfaces.

[routerVtep]	*table(string) = MemberInfo* see 2.576.1.1.1	List of router VTEPs.

2.576.1.1 GroupMembershipInfo

[interface]	*table(Interface name) = MemberInfo* see 2.576.1.1.1	Member ports information.
[vtep]	*table(string) = MemberInfo* see 2.576.1.1.1	Member VTEPs information.

2.576.1.1.1 MemberInfo

[mcastTunnelInfo]	*McastTunnelInfo* see 2.576.1.1.1.1	Multicast Tunnel Information.
[reporterInfo]	*table(IPv4 or IPv6 address) = ReporterInfo* see 2.576.1.1.1.2	Host information.

2.576.1.1.1.1 McastTunnelInfo

[tunnelType]	*pimTunnel*	Type of multicast tunnel.

2.576.1.1.1.2 ReporterInfo

[creationTime]	*floating point number*	The timestamp in UTC when first heard from host.
[expiryTime]	*floating point number*	The timestamp in UTC when membership will expire.
[lastHeard]	*floating point number*	The timestamp in UTC when last heard from host.

2.577 show ip igmp snooping groups count [...]

numberOfGroups	*integer*	Number of groups in membership.

2.578 show ip igmp snooping groups detail [...]

see 2.576

2.579 show ip igmp snooping groups proxy [...]

detail	*boolean*	Include detailed attributes.
vlanInfo	*table(numeric string) = IgmpSnoopingProxyPorts* see 2.579.1	Dict of VlanId to per interface proxy related information.

2.579.1 IgmpSnoopingProxyPorts

portInfo	*table(Interface name) = IgmpSnoopingProxyGroups* see 2.579.1.1	Dict of Interface to Groups that are reported.

2.579.1.1 IgmpSnoopingProxyGroups

groups	*table(IPv4 or IPv6 address) = IgmpSnoopingProxyGroupInfo* see 2.579.1.1.1	Map Multicast addresses to an optional sourceList.

2.579.1.1.1 IgmpSnoopingProxyGroupInfo

[sourceList]	*IgmpSnoopingProxySourceList* see 2.579.1.1.1.1	Optional Source List.

2.579.1.1.1.1 IgmpSnoopingProxySourceList

filterMode	*enum: Exclude | Include*	Source filter mode.
sources	*list of IP address*	List of sources associated with the multicast group.

2.580 show ip igmp snooping mrouter [...]

vlans	*table(numeric string) = IgmpSnoopingMrouterVlan* see 2.580.1	Maps a VLAN ID to VLAN information.

2.580.1 IgmpSnoopingMrouterVlan

interfaces	*list of IgmpSnoopingMrouterDetail* see 2.580.1.1	Mrouter interface info.

2.580.1.1 IgmpSnoopingMrouterDetail

[address]	*IP address*	Source address of the interface.
[expireTime]	*floating point number*	UTC time at which this router will expire.
[firstHeardTime]	*floating point number*	UTC time at which this router is first heard.
interfaceName	*interface name*	Multicast router interface.
[lastHeardTime]	*floating point number*	UTC time at which this router is last heard.
routerType	*enum: dynamic | unknown | mrd | static | pim | querier | dvmrp*	Type of multicast router.

2.581 show ip igmp snooping querier [...]

see 2.434

2.582 show ip igmp snooping querier counters [...]

see 2.435

2.583 show ip igmp snooping querier membership [...]

see 2.436

2.584 show ip igmp snooping querier membership vlan VLAN_ID group IPADDR

see 2.437

2.585 show ip igmp snooping querier status [...]

see 2.438

2.586 show ip igmp snooping report-flooding [...]

`[reportFloodingEnabled]`	*boolean*	True if ReportFlooding enabled.
`reportFloodingSwitchPorts`	*list of Interface name*	Interfaces for which report-flooding enabled.
`[vlans]`	*table(numeric string) = IgmpSnoopingVlanReportFlooding* see 2.586.1	Maps vlanId to vlan information.

2.586.1 IgmpSnoopingVlanReportFlooding

`[reportFloodingEnabled]`	*boolean*	True if report flooding enabled in vlan.
`reportFloodingOperationalState`	*enum: disabled | default | enabled*	State of report flooding in vlan.
`sanityCheckFailedPorts`	*list of Interface name*	Sanity check for all Switchports.
`switchPorts`	*table(Interface name) = string*	A mapping of ports forwarding IGMP membership report to port status.

2.587 show ip igmp static-groups [...]

`intfAddrs`	*table(Interface name) = IgmpStaticGroupsInterface* see 2.587.1	Map interface name to list of source-group addrs pairs.

2.587.1 IgmpStaticGroupsInterface

[aclInfo]	*table(string) = IgmpAclInfo* see 2.587.1.1	map from ACL names to ACL info objects.
groupAddrsList	*list of InterfaceAddrs* see 2.587.1.2	list of source-group addr pairs for this interface.

2.587.1.1 IgmpAclInfo

aclDetails	*list of AclInfo* see 2.587.1.1.1	list of objects containing detailed information about ACLs.
[errorMessage]	*string*	error message for improperly configured ACL.

2.587.1.1.1 AclInfo

[sequence]	*integer*	The sequence number for an ACL on an interface.
sourceGroupInfo	*string*	source and group addresses for this ACL or an error message.

2.587.1.2 InterfaceAddrs

groupAddr	*IPv4 address*	group address for group interface.
sourceAddr	*IPv4 address*	source address for group interface.

2.588 show ip igmp static-groups group [...]

sourceGroups	*table(IPv4 address) = IgmpInterfacesList* see 2.588.1	Map from sourceGroup addr to interface names.

2.588.1 IgmpInterfacesList

interfaces	*list of Interface name*	List of interface names.

2.589 show ip igmp statistics [...]

IgmpStatistics	*table(Interface name)* = *IgmpStatisticsInterface* see 2.589.1	Map of statistics for IGMP interfaces.

2.589.1 IgmpStatisticsInterface

errorPacketReceived	*integer*	error packets received.
otherPacketReceived	*integer*	other packets received.
totalGeneralQueriesSent	*integer*	IGMP total general queries sent.
v1QueriesReceived	*integer*	IGMP v1 queries received.
v1QueriesSent	*integer*	IGMP v1 queries sent.
v1ReportReceived	*integer*	IGMP v1 reports received.
v2LeaveReceived	*integer*	IGMP v2 leaves received.
v2QueriesReceived	*integer*	IGMP v2 queries received.
v2QueriesSent	*integer*	IGMP v2 queries sent.
v2ReportReceived	*integer*	IGMP v2 reports received.
v3GSQueriesSent	*integer*	IGMP v3 group-specific queries sent.
v3GSSQueriesSent	*integer*	IGMP v3 group-source-specific queries sent.
v3QueriesReceived	*integer*	IGMP v3 queries received.
v3QueriesSent	*integer*	IGMP v3 queries sent.
v3ReportReceived	*integer*	IGMP v3 reports received.

2.590 show ip interface [...]

revision 2

interfaces	*table(Interface name)* = *IpStatus* see 2.590.1	Maps an interface name to its IP status.

2.590.1 IpStatus

addresslessForwarding	*enum: isInvalid* | *isTrue* | *isFalse*	V4 Forwarding mode on the V6 interface.
[description]	*string*	Description of this interface.
directedBroadcastEnabled	*boolean*	Whether or not Directed Broadcast is enabled for this intf.
enabled	*boolean*	Whether or not this interface is enabled.
gratuitousArp **	*boolean*	Whether or not gratuitous ARP is accepted. **Since: 4.20.5
interfaceAddress	*InterfaceAddress* see 2.467.1.2	Network Layer Addresses for this interface.
[interfaceAddressBrief] **	*InterfaceAddressBrief* see 2.467.1.3	Network layer address for this interface. **Since: 4.24.2
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
[ipv4Routable0] **	*boolean*	Whether or not IPv4 routable 0.0.0.0/8is enabled. **Since: 4.29.2
[ipv4Routable240] **	*boolean*	Whether or not IPv4 routable 240.0.0.0/4is enabled. **Since: 4.24.2
[isVrrpBackup] **	*boolean*	Whether if the VRRP instance is in VRRP backup mode. **Since: **currentRelease**
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
localProxyArp	*boolean*	Whether or not local proxy-ARP is enabled.
[maxMssEgress] **	*integer*	TCP MSS ceiling in bytes enforced on packets forwarded to the network. **Since: 4.21.3
[maxMssIngress] **	*integer*	TCP MSS ceiling in bytes enforced on packets arriving from the network. **Since: 4.21.3
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
proxyArp	*boolean*	Whether or not proxy-ARP is enabled.

proxyArpAllowDefault **	*boolean*	Whether or not proxy-ARP always consider the default route if it is present in the routing table. **Since: **currentRelease**
[proxyArpVirtualMacAddress] **	*MAC address*	The virtual MAC address used in proxy-ARP or local proxy-ARP. **Since: **currentRelease**
[routedAddr] **	*MAC address*	The routed MAC address used in proxy-ARP or local proxy-ARP. **Since: **currentRelease**
[urpf]	*enum: strict* | *loose* | *strictDefault* | *disable* | *looseNonDefault*	Unicast Reverse Path Forwarding mode on the interface.
[vrId] **	*integer*	The VRRP group of the virtual router seleted in proxy ARP. **Since: **currentRelease**
[vrf]	*string*	Name of the VRF.

2.591 show ip interface brief [...]

interfaces	*table(Interface name) = IpStatusBrief* see 2.591.1	Maps an interface name to an abbreviated IP status.

2.591.1 IpStatusBrief

interfaceAddress	*InterfaceAddressBrief* see 2.467.1.3	Network layer address for this interace.
[interfaceAddressBrief] **	*InterfaceAddressBrief* see 2.467.1.3	Network layer address for this interface. **Since: 4.24.2
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
[ipv4Routable0] **	*boolean*	Whether or not IPv4 routable 0.0.0.0/8is enabled. **Since: 4.29.2
[ipv4Routable240] **	*boolean*	Whether or not IPv4 routable 240.0.0.0/4is enabled. **Since: 4.24.2
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.
[nonRoutableClassEIntf] **	*boolean*	True if class E address configured and class E address routing is disabled. **Since: 4.24.2

2.592 show ip large-community-list [...]

`ipLargeCommunityLists`	*table(string) = IpLargeCommunityList* see 2.592.1	Table of large community lists by name.

2.592.1 IpLargeCommunityList

`entries`	*list of IpCommunityListEntry* see 2.537.1.1	Large community list entries sorted by increasing sequence number.

2.593 show ip load-sharing trident fields

revision 2

ecmpHashFieldsCommon **	*EcmpHashFieldsCommon* see 2.593.1	ECMP hashing on MISC fields. **Since: 4.23.2
ecmpHashFieldsGtp **	*EcmpHashFieldsGtp* see 2.593.2	ECMP hashing on GTP fields. **Since: 4.23.1
ecmpHashFieldsIp	*EcmpHashFieldsIp* see 2.593.3	ECMP hashing on IP fields.
ecmpHashFieldsIpv6	*EcmpHashFieldsIpv6* see 2.593.4	Ecmp hashing on IPV6 fields.
hashSeed	*integer*	hash seed.

2.593.1 EcmpHashFieldsCommon

sysMacAddrHashed	*boolean*	ECMP hashing on the system MAC address.

2.593.2 EcmpHashFieldsGtp

gtpVersion	*string*	Hash on either GTP version 1 or 2.
teid	*boolean*	Ecmp hashing on GTP fields.

2.593.3 EcmpHashFieldsIp

Could also be an instance of:
EcmpHashFieldsIpv6 2.593.4

dstIpHashed	*boolean*	ECMP hashing on destination IP address.
dstIpPortHashed	*boolean*	ECMP hashing on destination IP port.
ingressIntfHashed	*boolean*	ECMP hashing on ingress interface.
srcIpHashed	*boolean*	ECMP hashing on source IP address.
srcIpPortHashed	*boolean*	ECMP hashing on source IP port.

2.593.4 EcmpHashFieldsIpv6

dstIpHashed	*boolean*	ECMP hashing on destination IP address.
dstIpPortHashed	*boolean*	ECMP hashing on destination IP port.
flowLabelHashed	*boolean*	ECMP hashing on flow label.
ingressIntfHashed	*boolean*	ECMP hashing on ingress interface.
srcIpHashed	*boolean*	ECMP hashing on source IP address.
srcIpPortHashed	*boolean*	ECMP hashing on source IP port.

2.594 show ip mfib PIM_MODE [...]

groups	*table(IPv4 or IPv6 address prefix) = MfibGroup* see 2.594.1	Map of multicast groups.
hwPollingInterval	*integer*	Hardware polling interval.
hwResourceFull	*boolean*	No space remaining in hardware.

2.594.1 MfibGroup

routes	*table(IPv4 or IPv6 address prefix) = MfibRoute* see 2.594.1.1	Multicast routes for this group.

2.594.1.1 MfibRoute

[bitsRateCounter] **	*RateCounter* see 2.594.1.1.3	Hardware multicast traffic counter rate, in bits/s, if supported. **Since: 4.26.0
[bytesCounter]	*Counter* see 2.594.1.1.1	Hardware multicast traffic counter, in bytes, if supported.
[fastdropInterfaces]	*list of Interface name*	List of fastdrop interfaces.
[hwRoute]	*MfibHwRoute* see 2.594.1.1.2	Hardware route information.
[incomingInterface]	*interface name*	Incoming interface for the route.
[incomingInterfaceFrr]	*interface name*	Incoming interface for fast failover route.
[notifyInterface]	*interface name*	Notify interface for the route.
[outgoingInterfaces]	*list of Interface name*	List of outgoing interfaces for the route.
[packetsCounter]	*Counter* see 2.594.1.1.1	Hardware multicast traffic counter, in packets, if supported.
[pktsRateCounter] **	*RateCounter* see 2.594.1.1.3	Hardware multicast traffic counter rate, in pkts/s, if supported. **Since: 4.26.0
[rpaIndex]	*integer*	Rendezvous point address index for the route.
[staticFastdropInterface]	*interface name*	Interface with static fastdrop installed.
[toCpu]	*boolean*	Route has local listeners.

2.594.1.1.1 Counter

[active]	*boolean*	Counter active in hardware.
[value]	*integer*	Stored count retrieved from hardware when counter is active.

2.594.1.1.2 MfibHwRoute

creationTime	*floating point number*	UTC time when the hardware route was created.
programmed	*boolean*	Route programmed in hardware.

[routeActivity]	*floating point number*	Last polled UTC time when the route was seen as active in hardware.

2.594.1.1.3 RateCounter

[active]	*boolean*	Counter active in hardware.
[value]	*floating point number*	Stored counter rate.

2.595 show ip mfib [...]

see 2.594

2.596 show ip mfib df [...]

bidirEnabled	*boolean*	Pim Bidir enabled.
dfProfiles	*table(Interface name) = integer*	Mapping from interface to DF bitmap.

2.597 show ip mfib rpa [...]

bidirEnabled	*boolean*	Pim Bidir enabled.
groups	*table(IPv4 or IPv6 address prefix) = integer*	Mapping from prefix to RPA index.

2.598 show ip mroute GROUP_OR_SOURCE1 interface [...]

revision 2

bidirectional	*MrouteInterfaceDetails* see 2.598.1	IP Mroute Interfaces for Bidirectional PIM.
sparseMode	*MrouteSmInterfaces* see 2.20	IP Mroute Interfaces for Sparse-mode PIM.

2.598.1 MrouteInterfaceDetails

interfaces	*table(Interface name) = MrouteInterface* see 2.598.1.1	Map of interfaces for multicast route.

2.598.1.1 MrouteInterface

[address]	*IP address*	Interface IP address.
downstreamJoinState	*enum: noInfo | prunePending | joined*	Downstream join state.
forward	*boolean*	Member of outgoing interface set.
iAmDf	*boolean*	This Pim router is the designated router on this interface.

[lastJoinReceived]	*floating point number*	UTC time at which the last downstream join was received.
localReceiverInclude	*boolean*	Local receiver include (*,G,I).

2.599 show ip mroute [...]

revision 2

bidirectional	*Groups* see 2.599.1	IP Mroute for Bidirectional PIM.
sparseMode	*GroupSms* see 2.21	IP Mroute for Sparse-mode PIM.

2.599.1 Groups

groups	*table(IPv4 or IPv6 address) = Mroute* see 2.599.1.1	Map of bidirectional mode PIM multicast groups.

2.599.1.1 Mroute

[active]	*boolean*	Group active.
creationTime	*floating point number*	UTC time at which the route was created.
[details]	*MrouteInterfaceDetails* see 2.598.1	Detailed information for this multicast route.
joinDesired	*boolean*	Join Desired (*, G).
oifList	*list of Interface name*	Outgoing interface list.
[rp]	*IP address*	RP(G): The Rendezvous Point for this group.
[rpf]	*Rpf* see 2.599.1.1.1	Reverse path forwarding information.
[rpfFrr] **	*Rpf* see 2.599.1.1.1	Reverse path forwarding information for MoFRR. **Since: 4.20.0
[rpfFrrInterface] **	*interface name*	Reverse path forwarding interface for MoFRR. **Since: 4.20.0
[rpfInterface]	*interface name*	Reverse path forwarding interface, packets for this mroute must arrive on this interface to be accepted/forwarded. If this value is set to 'Register', the RPF interface is the register tunnel.
upstreamJoinState	*enum: upNotJoined | upJoined*	Upstream joined state.

2.599.1.1.1 Rpf

rpfMetric	*integer*	RPF route metric.
[rpfNeighbor]	*IP address*	RPF neighbor.
rpfPreference	*integer*	RPF route preference.
rpfPrefix	*IP address prefix*	RPF route prefix.
rpfRib	*enum: M | none | U*	RPF routing table.

2.600 show ip mroute bidirectional GROUP_ADDR interface [...]

see 2.599.1

2.601 show ip mroute bidirectional [...]

see 2.599.1

2.602 show ip mroute bidirectional count [...]

groups	*integer*	Number of groups.

2.603 show ip mroute count [...]

revision 2

bidirectional	*MrouteBidirCount* see 2.602	count for bidirectional PIM.
sparseMode	*MrouteCount* see 2.22	count for Sparse-mode PIM.

2.604 show ip mroute sparse-mode (sources | groups) [...]

isSource	*boolean*	If true, we print S-> no of G.
sourceGroup	*table(numeric string) = SourceGroupCollection* see 2.604.1	S-> no of G or G-> no of S mapping.

2.604.1 SourceGroupCollection

sourceOrGroupAddrs	*table(IPv4 or IPv6 address) = string*	List of IP address.

2.605 show ip mroute vrf all [...]

vrfs	*table(string) = IpMroutesAll* see 2.599	Multicast routes for all VRFs.

2.606 show ip msdp mesh-group [...]

members	*table(string) = Group* see 2.606.1	Dictionary of Mesh group names to list of member IPs.

2.606.1 Group

ipList	*list of IPv4 address*	Mesh group members ip.

2.607 show ip msdp peer [...]

revision 2

peerList	*list of PeerInfo* see 2.607.1	list of peers.

2.607.1 PeerInfo

acceptedSas	*list of SaMsg* see 2.607.1.1	List of Accepted SAs.
[connSourceAddress]	*IPv4 address*	The connection source IP address, If Available.
[connSourceInterface]	*interface name*	Connection source interface used, If available.
[description]	*string*	Description of MSDP peer.
peerIpAddress	*IPv4 address*	Ip address of the MSDP peer.
resetCount	*integer*	MSDP peer connection reset count.
saCount	*integer*	Number of SA messages from this MSDP peer in the SA cache.
saFilterIn	*string*	Access list filtering of SA input.
saFilterOut	*string*	Access list filtering of SA output.
[sessionStartTime]	*floating point number*	UTC time at which the peer last switched state to established.
state	*enum: inactive | established | disabled | connecting | listen*	Current status of the peer.

2.607.1.1 SaMsg

[remoteAddress]	*IPv4 address*	Heard from this peer.
rpAddress	*IPv4 address*	Addess of the Rendezvous Point.
sourceGroupPair	*SGPair* see 2.607.1.1.1	(S,G) Pair.

2.607.1.1.1 SGPair

groupAddress	*IPv4 address*	Group Address.
sourceAddress	*IPv4 address*	Source Address.

2.608 show ip msdp pim sa-cache [...]

saCache	*list of SaMsg* see 2.607.1.1	MSDP soure active messages for local Pim RP.

2.609 show ip msdp rpf-peer RP [...]

queryCompleted	*boolean*	If the RPF query completed.
queryStarted	*boolean*	RPF query started.
rpAddress	*IPv4 address*	RP address.

[rpfPeer]	*IPv4 address*	RPF peer address, if query succeeds.

2.610 show ip msdp sa-cache [...]

acceptedSaMsg	*list of SaMsg* see 2.607.1.1	List of accepted SA Messages.
[rejectedSaMsg]	*list of SaMsg* see 2.607.1.1	List of rejected SA Messages.

2.611 show ip msdp summary [...]

revision 2

peerList	*list of PeerSummary* see 2.611.1	list of peers.

2.611.1 PeerSummary

Could also be an instance of:
PeerInfo 2.607.1

[description]	*string*	Description of MSDP peer.
peerIpAddress	*IPv4 address*	Ip address of the MSDP peer.
resetCount	*integer*	MSDP peer connection reset count.
saCount	*integer*	Number of SA messages from this MSDP peer in the SA cache.
[sessionStartTime]	*floating point number*	UTC time at which the peer last switched state to established.
state	*enum: inactive | established | disabled | connecting | listen*	Current status of the peer.

2.612 show ip multicast boundary [...]

mcastBoundaryInterfaces	*table(Interface name) = McastBoundaryInterface* see 2.612.1	Map of interface to multicast boundary status.

2.612.1 McastBoundaryInterface

boundary	*table(IPv4 or IPv6 address prefix) = boolean*	Map of denied prefix to its data plane filtering status.

2.613 show ip name-server

nameServerConfigs **	*list of NameServerConfig* see 2.422.1	List of configured name servers. **Since: 4.26.0
v4NameServers	*list of string*	List of IPV4 name servers.
v6NameServers	*list of string*	List of IPV6 name servers.

2.614 show ip nat ((access-list | acl)) [...]

aclList	*list of NatAcl* see 2.614.1	List of the ACLs to display.

2.614.1 NatAcl

aclInfo **	*table(string) = NatAclInfo* see 2.614.1.2	NAT configuration for the ACL. **Since: 4.24.1
aclName	*string*	The name of the specified ACL.
aclRules	*list of AclRule* see 2.614.1.1	Rules that belongs to this ACL, ordered by the sequence number of the rule.
interfaces	*list of Interface name*	List of interfaces using this ACL for NAT.
profiles **	*list of string*	List of profiles using this ACL for NAT. **Since: 4.23.2

2.614.1.1 AclRule

description	*string*	Textual description of the ACL rule if valid, or explanation of why the rule doesn't apply if it isn't.
seqNo	*integer*	Sequence number of the rule.
valid	*boolean*	True if NAT can use this rule to filter translations.

2.614.1.2 NatAclInfo

addrOnly	*boolean*	Access list used in address only NAT.
fullCone	*boolean*	Access list used in full cone NAT.
poolId	*integer*	NAT pool ID associated with the access list.

2.615 show ip nat counters [...]

revision 2

vrfs	*table(string) = NatCounters* see 2.615.1	NAT counters of each VRF.

2.615.1 NatCounters

deleteConnectionErrors	*integer*	Conntrack delete connection errors.
deleteConnectionEvents	*integer*	Number of Conntrack delete connection events.
[eagainDumpCnt]	*integer*	Number of Netlink no data errors during kernel sync.
[eagainRecvCnt]	*integer*	Number of Netlink no data errors during kernel update.
[enobufDumpCnt]	*integer*	Number of Netlink no space errors during kernel sync.
[enobufMsgCnt]	*integer*	Number of Netlink no space errors during kernel update.
iptablesWriteErrors	*integer*	Kernel iptables write errors.
kernelSyncInProgress	*integer*	Conntrack table sync already in progress count.
kernelSyncRequestErrors	*integer*	Conntrack table sync request errors.
kernelSyncRequests	*integer*	Conntrack table sync requests.
[maxMsgCntInRecvBuf]	*integer*	Max number of messages processed in one buffer.
netlinkBadMessageLength-Errors	*integer*	Number of Netlink bad message length errors.
netlinkDumpDoneErrors	*integer*	Number of Netlink unexpected dump done messages.
netlinkDumpIgnoreTcp-Connections	*integer*	Number of Netlink TCP connections ignored in dump.
[netlinkMsgErrors]	*integer*	Number of Netlink msg errors.
netlinkNoBufferErrors	*integer*	Number of Netlink no space errors.
netlinkNoSpaceErrors	*integer*	Number of Netlink no space in message errors.
netlinkReceiveErrors	*integer*	Number of Netlink receive errors.
netlinkSeqNumMismatches	*integer*	Number of Netlink sequence number mismatches.
netlinkSeqNumRange-Mismatches	*integer*	Number of Netlink sequence number out of range errors.
netlinkSyncCreateTxErrors **	*integer*	Number of Netlink Sync creation event errors. **Since: 4.25.0
netlinkSyncDeleteTxErrors **	*integer*	Number of Netlink Sync deletion event errors. **Since: 4.25.0
netlinkSyncEagainErrors **	*integer*	Number of Netlink Sync EAGAIN errors. **Since: 4.25.0

netlinkSyncFCConflictErrors **	*integer*	Number of Netlink Sync full cone conflict event errors. **Since: 4.30.1
netlinkSyncMsgErrors **	*integer*	Number of Netlink Sync messages with errors. **Since: 4.25.0
netlinkSyncNoBufferErrors **	*integer*	Number of Netlink Sync ENOBUFS errors. **Since: 4.25.0
netlinkSyncReceiveErrors **	*integer*	Number of Netlink Sync receive errors. **Since: 4.25.0
netlinkSyncUpdateTxErrors **	*integer*	Number of Netlink Sync update event errors. **Since: 4.25.0
netlinkTruncatedMessages	*integer*	Number of Netlink truncated messages.
newConnectionEvents	*integer*	Number of Conntrack new connection events.
[nlAdd300s] **	*integer*	300s NAT entries learning rate. **Since: 4.27.1
[nlAdd30s] **	*integer*	30s NAT entries learning rate. **Since: 4.27.1
[nlAdd5s] **	*integer*	5s NAT entries learning rate. **Since: 4.27.1
[nlAdd60s] **	*integer*	60s NAT entries learning rate. **Since: 4.27.1
[nlDel300s] **	*integer*	300s NAT entries deletion rate. **Since: 4.27.1
[nlDel30s] **	*integer*	30s NAT entries deletion rate. **Since: 4.27.1
[nlDel5s] **	*integer*	5s NAT entries deletion rate. **Since: 4.27.1
[nlDel60s] **	*integer*	60s NAT entries deletion rate. **Since: 4.27.1
updateConnectionErrors	*integer*	Conntrack update connection errors.
updateConnectionEvents	*integer*	Number of Conntrack update connection events.

2.616 show ip nat flow action [...]

policies	*table(string) = NatFlowActionPolicy* see 2.616.1	NAT flow policies, keyed by name.

2.616.1 NatFlowActionPolicy

matches	*table(string) = NatFlowMatchActions* see 2.616.1.1	Matches for this policy, keyed by match name.
[profiles]	*list of string*	List of profiles using this NAT flow policy.

2.616.1.1 NatFlowMatchActions

actions	*table(string) = NatFlowAction* see 2.616.1.1.1	Actions for this match, keyed by action name.

2.616.1.1.1 NatFlowAction

[flowType]	*string*	NAT flow type.
intfs	*table(Interface name) = NatFlowIntf* see 2.616.1.1.1.2	Flow outputs, keyed by interface.
natDst	*IpAddrAndPort* see 2.616.1.1.1.1	The destination IP address and port for the NAT translation.
natSrc	*IpAddrAndPort* see 2.616.1.1.1.1	The source IP address and port for the NAT translation.

2.616.1.1.1.1 IpAddrAndPort

ip	*IPv4 address*	Ip address.
port	*integer*	Port.

2.616.1.1.1.2 NatFlowIntf

mac	*MAC address*	The destination MAC address for the output.
vlanId	*integer*	The VLAN ID for the output.

2.617 show ip nat flow match [...]

hardware	*boolean*	NAT flow hardware state.
policies	*table(string) = NatFlowMatchPolicy* see 2.617.1	NAT flow policies, keyed by name.

2.617.1 NatFlowMatchPolicy

matches	*table(string) = NatFlowMatch* see 2.617.1.1	Matches for this policy, keyed by match name.

2.617.1.1 NatFlowMatch

flow	*NatFlowAddress* see 2.617.1.1.1	NAT flow match fields.
[mcGroup]	*integer*	Hardware multicast replication group.
[ucGroup]	*integer*	Hardware unicast replication group.

2.617.1.1.1 NatFlowAddress

dstAddr	*IpAddrAndMask* see 2.241.1.2.2.9	The destination IP address to match on.
dstPortEnd	*integer*	The end of the destination port range of the match.
dstPortStart	*integer*	The start of the destination port range of the match.
priority	*integer*	The priority of the match.
protocol	*integer*	The protocol of the match.
srcAddr	*IpAddrAndMask* see 2.241.1.2.2.9	The source IP address to match on.
srcPortEnd	*integer*	The end of the source port range of the match.
srcPortStart	*integer*	The start of the source port range of the match.

2.618 show ip nat flow profile [...]

hardware	*boolean*	NAT flow hardware state.
profiles	*table(string) = NatFlowProfilePolicy* see 2.618.1	NAT flow profiles, keyed by name.

2.618.1 NatFlowProfilePolicy

policies	*table(string) = NatFlowPolicyInfo* see 2.618.1.1	NAT flow policy details, keyed by policy name.

2.618.1.1 NatFlowPolicyInfo

[match]	*table(string) = NatFlowMatchInfo* see 2.618.1.1.1	NAT flow match details, keyed by match name.

2.618.1.1.1 NatFlowMatchInfo

groupMc	*integer*	Multicast replication group used for multicast traffic.
groupUc	*integer*	Multicast replication group used for unicast traffic.

2.619 show ip nat logging

enabled	*boolean*	NAT event logging is enabled.

2.620 show ip nat pool [...]

[natPools]	*table(string) = NatPool* see 2.620.1	Map of NAT pools keyed by their name.
[poolName]	*string*	Name of the NAT pool to show.
[showSplit] **	*boolean*	Show local NAT sync port range split. **Since: 4.29.2

2.620.1 NatPool

natRanges	*list of NatRange* see 2.620.1.1	List of NAT pool ranges.
poolId **	*integer*	NAT pool ID. **Since: 4.24.1
prefixLen	*integer*	Length of the prefix.

2.620.1.1 NatRange

endIp	*IPv4 address*	NAT end IP address.
endPort	*integer*	NAT end port number.
[localEndPort] **	*integer*	NAT sync port split end. **Since: 4.29.2
[localStartPort] **	*integer*	NAT sync port split start. **Since: 4.29.2
startIp	*IPv4 address*	NAT start IP address.
startPort	*integer*	NAT start port number.

2.621 show ip nat profile [...]

profiles	*list of NatProfile* see 2.621.1	List of profiles.

2.621.1 NatProfile

[intfList]	*table(Interface name) = integer*	List of interfaces configured with the NAT profile.
profileId	*integer*	The numeric ID assigned to each NAT profile.
profileName	*string*	The name of the specified NAT profile.
vrfName **	*string*	The VRF associated with each NAT profile. **Since: 4.23.1

2.622 show ip nat service-list [...]

serviceLists	*list of NatServiceList* see 2.622.1	List of service-list.

2.622.1 NatServiceList

serviceCmdList	*list of NatServiceList-CommandModel* see 2.622.1.1	List of service commands.
serviceListName	*string*	The name of the specified service-list.

2.622.1.1 NatServiceListCommandModel

protocol	*string*	protocol for ALG.

2.623 show ip nat synchronization

connAttempts	*integer*	Number of attempts at synchronizing with peer.
connEstablishedTime	*floating point number*	Peer Connection Establishment time.
connPort	*integer*	Port used for the connection.
connSrcIp	*IPv4 address*	Connection source Ip.
connState	*enum: disconnected* | *connected* | *connecting* | *init* | *disconnecting* | *failed*	State of the connection with peer.
fwdToPeer	*table(string) = NatSyncFwdToPeer* see 2.623.2	NAT profiles' fowardTopeer status.
kernelDevName	*string*	Kernel device name.
localIntf	*interface name*	Local interface.
natVerCompatible	*boolean*	Peer's version is Compatible.
natVersion	*Version* see 2.623.1	Nat version.
peerIpAddr	*IPv4 address*	Peer's IP address.
portRangeMode	*enum: portRangeFull* | *portRangeLower* | *portRangeUpper*	Port range mode used in the system.
shutdown	*boolean*	Shutdown StateFalse: peer connection is administratively down, True: peer connection is administratively up.
statusMountState	*enum: mountUnmounted* | *mountMounted* | *mountMounting* | *mountFailed*	Mount state of peer's status.
versionMountState	*enum: mountUnmounted* | *mountMounted* | *mountMounting* | *mountFailed*	Mount state of peer's version.

2.623.1 Version

maxVersion	*integer*	Maximum version number.
minVersion	*integer*	Minimum version number.

2.623.2 NatSyncFwdToPeer

peerIntf	*interface name*	Local interface connected to peer.
peerIpAddr	*IPv4 address*	Peer's IP address.
peerResolved	*boolean*	Peer's reachability status.

2.624 show ip nat synchronization advertised-translations [...]

revision 3

detailedView	*boolean*	includes additional details if True.
translations	*list of Translation* see 2.624.1	List of NAT translations active on this interface.

2.624.1 Translation

[destinationAddress]	*IpAddr* see 2.624.1.1	The destination IP address and port/mask of the packet.
[details]	*Details* see 2.624.1.2	Details, if requested.
globalAddress	*IpAddr* see 2.624.1.1	The translated IP address and port/mask of the packet.
[interface]	*string*	Interface or NAT profile name.
natType	*enum: STAT | D-TW | DYN | FC | AO*	Static NAT or Dynamic NAT.
[sourceAddress]	*IpAddr* see 2.624.1.1	The original source IP address and port/mask of the packet.
target	*enum: DST | SRC*	Source NAT or destination NAT.

2.624.1.1 IpAddr

Could also be an instance of:
IpAddrAndMask 2.241.1.2.2.9
IpAddrAndPort 2.616.1.1.1.1

ip	*IPv4 address*	Ip address.

2.624.1.2 Details

lastActivityTime	*floating point number*	Last time data packets matching this translation were seen.
packetCount	*integer*	Number of packets translated from the original IP address.
packetReplyCount	*integer*	Number of packets translated to the original IP address.
[protocol]	*enum: ICMP | TCP | IGMP | UDP*	IP Protocol, if specified.
[protocolNumber]	*integer*	IP Protocol Number, if specified.
[sourceVrfName]	*string*	Source VRF name.
[tcpState]	*enum: SYN_RECV | FIN_WAIT | LAST_ACK | TIME_WAIT | LISTEN | RETRANS | TIMEOUT_MAX | NONE | CLOSE_WAIT | SYN_SENT | CLOSE | IGNORE | ESTABLISHED | UNACK | MAX*	State of the TCP connection.

vrfName	*string*	VRF name.

2.625 show ip nat synchronization discovered-translations [...]

revision 3

see 2.624

2.626 show ip nat synchronization peer

revision 2

connAttempts	*integer*	Number of attempts at synchronizing with peer.
connEstablishedTime	*floating point number*	Peer Connection Establishment time.
connPort	*integer*	Port used for the connection.
connSrcIp	*IPv4 address*	Connection source Ip.
connState	*enum: disconnected* | *connected* | *connecting* | *init* | *disconnecting* | *failed*	State of the connection with peer.
kernelDevName	*string*	Kernel device name.
localIntf	*interface name*	Local interface.
natVerCompatible	*boolean*	Peer's version is Compatible.
natVersion	*Version* see 2.623.1	Nat version.
peerIpAddr	*IPv4 address*	Peer's IP address.
portRangeMode **	*enum: portRangeFull* | *portRangeLower* | *portRangeUpper*	Port range mode used in the system. **Since: 4.28.2
shutdown	*boolean*	Shutdown StateFalse: peer connection is administratively downTrue: peer connection is administratively up.
statusMountState	*enum: mountUnmounted* | *mountMounted* | *mountMounting* | *mountFailed*	Mount state of peer's status.
versionMountState	*enum: mountUnmounted* | *mountMounted* | *mountMounting* | *mountFailed*	Mount state of peer's version.

2.627 show ip nat translation (address-only | full-cone) flows [...]

intfTranslationFlows	*table(string) = IntfTranslationFlows* see 2.627.1	List of active NAT translation flows per interface.

2.627.1 IntfTranslationFlows

translationFlows	*list of TranslationFlows* see 2.627.1.1	Flows using an address-only or full-cone NAT Translation.
vrfName	*string*	VRF name.

2.627.1.1 TranslationFlows

flows	*list of TranslationFlow* see 2.627.1.1.1	List of active NAT translations.
globalAddr	*IpAddrAndPort* see 2.616.1.1.1.1	NAT address.

2.627.1.1.1 TranslationFlow

destination	*IpAddrAndPort* see 2.616.1.1.1.1	Packet flow destination IP address and L4 port.
established	*boolean*	Network connection established.
protocol	*enum: IGMP* | *UDP* | *ICMP* | - | *TCP*	IP Protocol.
source	*IpAddrAndPort* see 2.616.1.1.1.1	Packet flow source IP address and L4 port.
[sourceVrfName] **	*string*	Source VRF name. **Since: 4.28.0
target	*enum: DST* | *SRC*	Source NAT or destination NAT.

2.628 show ip nat translation DYNAMIC_TYPE [...]

revision 3

detailedView	*boolean*	includes additional details if True.
hasComments **	*boolean*	True if any translations shown have comments. **Since: 4.28.1
translations	*list of Translation* see 2.628.1	List of NAT translations active on this interface.

2.628.1 Translation

[comment] **	*string*	comment on translation rule. **Since: 4.28.1
[details]	*Details* see 2.628.1.1	Details, if requested.
[direction] **	*enum: EGR | ING*	Ingress or Egress interface. **Since: 4.25.1
[dstAddress]	*IpAddr* see 2.624.1.1	The destination IP address and port/mask of the packet.
[interface]	*string*	Interface or NAT profile name.
natType	*enum: D-TW | DYN | ST | FC | AO | E-ST | TW*	Static NAT or Dynamic NAT.
[newDstAddress]	*IpAddr* see 2.624.1.1	The translated destination IP address and port/mask of the packet.
[newSrcAddress]	*IpAddr* see 2.624.1.1	The translated source IP address and port/mask of the packet.
[protocol]	*integer*	IP Protocol Number, if specified.
[srcAddress]	*IpAddr* see 2.624.1.1	The original source IP address and port/mask of the packet.
[target]	*enum: DST | SRC*	Source NAT or destination NAT.

2.628.1.1 Details

[established] **	*boolean*	Network connection established. **Since: 4.26.2
origin **	*string*	Discovered locally or advertised from sync peer. **Since: **currentRelease**
packetCount	*integer*	Number of packets translated from the original IP address.
packetReplyCount	*integer*	Number of packets translated to the original IP address.
[sourceVrfName] **	*string*	Source VRF name. **Since: 4.28.0
vrfName	*string*	VRF name.

2.629 show ip nat translation STATIC_TYPE [...]

revision 3

see 2.628

2.630 show ip nat translation [...]

revision 3

see 2.628

2.631 show ip nat translation address-only mapping [...]

revision 2

addrOnlyTranslations	*table(string) = SrcIpTranslations* see 2.631.1	List of active Nat translations per interface.

2.631.1 SrcIpTranslations

srcIpTranslations	*table(IPv4 address) = TranslationList* see 2.631.1.1	Translations for a source Ip.

2.631.1.1 TranslationList

translations	*list of Translation* see 2.631.1.1.1	List of active NAT translations.

2.631.1.1.1 Translation

[destinationAddress]	*IpAddrAndPort* see 2.616.1.1.1.1	The destination IP address and port/mask of the packet.
globalAddress	*IpAddrAndPort* see 2.616.1.1.1.1	The translated IP address and port/mask of the packet.
natType	*enum: STAT | DYN*	Static NAT or Dynamic NAT.
protocol	*enum: IGMP | UDP | ICMP | - | TCP*	IP Protocol.
[sourceAddress]	*IpAddrAndPort* see 2.616.1.1.1.1	The original source IP address and port/mask of the packet.
target	*enum: DST | SRC*	Source NAT or destination NAT.
vrfName **	*string*	VRF name. **Since: 4.23.2

2.632 show ip nat translation connection limit drop [...]

`vrfs`	*table(string) = DropTable* see 2.632.1	Connection drop counters, keyed by VRF name.

2.632.1 DropTable

`counters`	*list of DropCounter* see 2.632.1.1	Counters.
`vrfName`	*string*	VRF name.

2.632.1.1 DropCounter

`counter`	*integer*	Counter.
`dstIp`	*IpAddr* see 2.624.1.1	Destination IP.
`srcIp`	*IpAddr* see 2.624.1.1	Source IP.

2.633 show ip nat translation max-entries

[allHostsConnLimit]	*integer*	Hosts' connection limit.
[allHostsLowMarkPerc]	*integer*	Below this percentage, connection drops are disabled.
[allPoolsLowMarkPerc] **	*integer*	Below this percentage, connection drops are disabled. **Since: 4.29.1
connectionLimit	*integer*	Global connection limit.
connectionLimitLowMark	*integer*	Below this count, connection drops are disabled.
connectionLimitLowMarkPerc	*integer*	Below this percentage, connection drops are disabled.
fullConeConnLimit **	*integer*	Full-cone connection limit. **Since: 4.29.2
fullConeConnLimitLowMark **	*integer*	Below this count, full-cone connection drops are disabled. **Since: 4.29.2
fullConeConnLimitLowMark-Perc **	*integer*	Below this percentage, full-cone connection drops are disabled. **Since: 4.29.2
[hostConnLimits]	*list of NatHostConnectionLimit* see 2.633.1	List of host's connection limit.
numberOfFullConeConnections **	*integer*	Active full-cone connections in Nat agent. **Since: 4.29.2
numberOfSymmetric-Connections **	*integer*	Active symmetric connections in Nat agent. **Since: 4.29.2
[poolConnLimits] **	*list of NatPoolConnectionLimit* see 2.633.2	List of pool's connection limit. **Since: 4.29.1
symmetricConnLimit **	*integer*	Symmetric connection limit. **Since: 4.29.2
symmetricConnLimitLowMark **	*integer*	Below this count, symmetric connection drops are disabled. **Since: 4.29.2
symmetricConnLimitLowMark-Perc **	*integer*	Below this percentage, symmetric connection drops are disabled. **Since: 4.29.2
totalNumberOfConnections	*integer*	Active connections in Nat agent.
totalNumberOfLocal-Connections **	*integer*	Local connections in Nat Sync agent. **Since: 4.28.2

2.633.1 NatHostConnectionLimit

connectionLimit	*integer*	Connection limit per host.
hostIp	*IPv4 address*	Host IP address.
lowMarkPerc	*integer*	Below this percentage, connection drops are disabled.
[numberOfConnections]	*integer*	Number of connections per host.

[numberOfLocalConnections] **	*integer*	Number of connections per host in Nat Sync agent. **Since: 4.28.2

2.633.2 NatPoolConnectionLimit

connectionLimit	*integer*	Connection limit per pool.
lowMarkPerc	*integer*	Below this percentage, connection drops are disabled.
[numberOfConnections]	*integer*	Number of connections per pool.
[numberOfLocalConnections]	*integer*	Number of connections per pool in Nat Sync agent.
poolName	*string*	Pool name.

2.634 show ip nat translation max-entries IPADDR

see 2.633.1

2.635 show ip nat translation max-entries pool POOL

see 2.633.1

2.636 show ip nat translation rates

rates	*table(string) = NatTranslationRates* see 2.636.1	NAT translation learning rates.

2.636.1 NatTranslationRates

rate	*list of NatTranslationRate* see 2.636.1.1	NAT translations measured in a specific interval.

2.636.1.1 NatTranslationRate

numAdd	*integer*	Number of NAT entries added.
numDel	*integer*	Number of NAT entries deleted.
timeLen	*integer*	Length of rates measure.

2.637 show ip nat translation summary [...]

revision 3

translations	*table(string) = TranslationSummary* see 2.637.1	Map of NAT translation summary per interface.

2.637.1 TranslationSummary

numDynHwConnections	*integer*	Number of Dynamic NAT connections in Hardware.
numDynSwConnections	*integer*	Number of Dynamic NAT connections in Software.
numDynTwiceHwConnections	*integer*	Number of Dynamic twice NAT connections in Hardware.
numDynTwiceKernel-Connections	*integer*	Number of Dynamic twice NAT connections in Kernel.
numDynTwiceSwConnections	*integer*	Number of Dynamic twice NAT connections in Software.
numKernelConnections	*integer*	Number of Dynamic NAT connections in Kernel.
numStaticHwConnections	*integer*	Number of Static NAT connections in Hardware.
numStaticSwConnections	*integer*	Number of Static NAT connections in Software.
numTwiceHwConnections	*integer*	Number of Twice NAT connections in Hardware.
numTwiceSwConnections	*integer*	Number of Twice NAT connections in Software.

2.638 show ip nat translation twice [...]

revision 2

detailedView	*boolean*	includes additional details if True.
hasComments **	*boolean*	True if any translations shown have comments. **Since: 4.28.1
pendingView **	*boolean*	Excludes unnecessary columns if True. **Since: 4.24.0
translations	*list of Translation* see 2.638.1	List of Twice NAT translations active on this interface.

2.638.1 Translation

[comment] **	*string*	Comment on translation rule. **Since: 4.28.1
[details]	*Details* see 2.638.1.1	Details, if requested.
dstIpAndPortNew	*IpAddrAndPort* see 2.616.1.1.1.1	The translated destination IP address and port of the packet.
dstIpAndPortOld	*IpAddrAndPort* see 2.616.1.1.1.1	The original destination IP address and port of the packet.
srcIpAndPortNew	*IpAddrAndPort* see 2.616.1.1.1.1	The translated source IP address and port of the packet.
srcIpAndPortOld	*IpAddrAndPort* see 2.616.1.1.1.1	The original source IP address and port of the packet.

2.638.1.1 Details

group	*integer*	Twice NAT Group.
[interface]	*string*	Interface/Profile.
origin **	*string*	Discovered locally or advertised from sync peer. **Since: **currentRelease**
packetCount	*integer*	Number of packets translated from the original IP address.
packetReplyCount	*integer*	Number of packets translated to the original IP address.
[protocol]	*enum: ICMP | TCP | IGMP | UDP*	IP Protocol, if specified.
[protocolNumber]	*integer*	IP Protocol Number, if specified.
vrfName **	*string*	VRF name. **Since: 4.23.2

2.639 show ip nat translation vrf [...]

detailedView	*boolean*	includes additional details if True.
pendingView **	*boolean*	Excludes unnecessary columns if True. **Since: 4.24.0

translations	*list of Translation* see 2.639.1	List of Vrf NAT translations active on this interface.

2.639.1 Translation

[destinationAddress]	*IPv4 address*	The destination IP address and port/mask of the packet.
[details]	*Details* see 2.639.1.1	Details, if requested.
globalAddress	*IPv4 address*	The translated IP address and port/mask of the packet.
insideVrf	*string*	Inside vrf name.
[sourceAddress]	*IPv4 address*	The original source IP address and port/mask of the packet.
target	*enum: DST | SRC*	Source NAT or destination NAT.

2.639.1.1 Details

[outsideVrf]	*string*	Outside vrf name.
packetCount	*integer*	Number of packets translated from the original IP address.
packetReplyCount	*integer*	Number of packets translated to the original IP address.

2.640 show ip ospf ((retransmission queue) | retransmission-list) [...]

vrfs	*table(string) = InstListOspfLsaRetransmissionList* see 2.640.1	OSPF LSA retransmission information for all VRFs.

2.640.1 InstListOspfLsaRetransmissionList

instList	*table(numeric string) = OspfLsaRetransmissionList* see 2.640.1.1	OSPF Instance list keyed by instance id.

2.640.1.1 OspfLsaRetransmissionList

neighbors	*table(IPv4 address) = OspfLsaRetransmissionInstance* see 2.640.1.1.1	Dictionary of retransmission list keyed by the neighbor's router ID.

2.640.1.1.1 OspfLsaRetransmissionInstance

interfaceAddress	*IPv4 address*	OSPF neighbor interface address.
interfaceName	*string*	OSPF neighbor interface name.
[nextRetransmissionTime]	*floating point number*	Next retransmission time in UTC.
retransmissions	*list of OspfLsaEntry* see 2.640.1.1.1.1	LSA retransmission list.
retransmitQueueLength	*integer*	LSA retransmission queue length.

2.640.1.1.1.1 OspfLsaEntry

advertisingRouter	*IPv4 address*	LSA advertising router.
age	*integer*	LSA age in seconds.
checksum	*integer*	LSA checksum.
linkStateId	*IPv4 address*	LSA link state identifier.
lsaType	*enum: routerLsa* | *asExternalLsa* | *summaryAsbrLsa* | *opaqueType11Lsa* | *summaryLsa* | *opaqueType9Lsa* | *opaqueType10Lsa* | *networkLsa* | *nssaLsa*	LSA type.
sequenceNumber	*integer*	LSA sequence number.

2.641 show ip ospf (request-list | (request queue)) [...]

vrfs	*table(string)* = *InstListOspfLsaRequestList* see 2.641.1	OSPF request list information for all VRFs.

2.641.1 InstListOspfLsaRequestList

instList	*table(numeric string)* = *OspfLsaRequestList* see 2.641.1.1	OSPF Instance list keyed by instance id.

2.641.1.1 OspfLsaRequestList

neighbors	*table(IPv4 address)* = *OspfLsaRequestInstance* see 2.641.1.1.1	Dictionary of request list keyed by the neighbor's router ID.

2.641.1.1.1 OspfLsaRequestInstance

interfaceAddress	*IPv4 address*	OSPF neighbor interface address.
interfaceName	*string*	OSPF neighbor interface name.
requestQueueLength	*integer*	LSA request queue length.
requests	*list of OspfLsaEntry* see 2.640.1.1.1.1	LSA request list.

2.642 show ip ospf [...]

vrfs	*table(string) = InstListOspfShowInstModel* see 2.642.1	OSPF instance information for all VRFs.

2.642.1 InstListOspfShowInstModel

instList	*table(numeric string) = OspfShowInstModel* see 2.642.1.1	OSPF Instance list keyed by instance id.

2.642.1.1 OspfShowInstModel

abr	*boolean*	This OSPF instance is an Area Border Router.
[adjacencyExchangeStart-Threshold]	*integer*	Adjacency exchange-start threshold.
[areaList]	*table(IPv4 address) = OspfShowInstAreaModel* see 2.642.1.1.2	List of areas in this OSPF instance.
asbr	*boolean*	This OSPF instance is an AS Border Router.
[ecmpMaximumNexthops]	*integer*	Maximum number of ECMP nexthops.
[externalLsaInformation]	*OspfShowInstExternalLsa-InfoModel* see 2.642.1.1.3	External LSA information.
floodPacing	*integer*	Flood packet pacing timer in msecs.
[gracefulRestartInfo] **	*OspfShowInstGraceful-RestartInfoModel* see 2.642.1.1.4	Graceful restart information. **Since: 4.15.3
instanceId	*integer*	OSPF instance ID.
[lsaInformation]	*OspfShowInstLsaInfoModel* see 2.642.1.1.5	LSA information.
lsaRetransmissionThreshold **	*integer*	LSA retransmission threshold. **Since: 4.18.0
maxLsaInformation	*OspfShowInstMaxLsaInfo-Model* see 2.642.1.1.6	Max LSA information.
[maxMetricExt]	*integer*	Max metric for external LSA.
[maxMetricIncStub]	*integer*	Max metric including stubs.
[maxMetricSum]	*integer*	Max metric for summary LSA.
[maxMetricTypeInformation]	*OspfShowInstMaxMetric-TypeInfoModel* see 2.642.1.1.7	Max metric type information.
[maxMetricUnsetInformation]	*OspfShowInstMaxMetric-UnsetInfoModel* see 2.642.1.1.8	Max metric unset information.
numAreas	*integer*	Number of areas.
numBackboneNeighbors **	*integer*	Number of backbone neighbors. **Since: 4.18.1
numNormalAreas	*integer*	Number of normal areas.
numNssaAreas	*integer*	Number of nssa areas.

numStubAreas	*integer*	Number of stub areas.
[opaqueLsaInformation]	*OspfShowInstOpaqueLsaInfoModel* see 2.642.1.1.9	Opaque LSA information.
[primaryDomainId] **	*string*	Primary domain identifier. **Since: 4.29.1
[referenceBandwidth] **	*integer*	Auto-cost reference bandwidth in Mbps. **Since: 4.15.3
[routerId]	*IPv4 address*	OSPF Router id.
[secondaryDomainIds] **	*list of OspfSecondaryDomainId* see 2.642.1.1.1	List of secondary domain identifiers. **Since: 4.29.1
shutDown **	*boolean*	Indicates if the OSPFv2 instance is currently shut down. **Since: 4.18.1
[spfInformation]	*OspfShowInstSpfInfoModel* see 2.642.1.1.10	SPF information.
[teRouterId] **	*IPv4 address*	OSPF traffic engineering router ID. **Since: 4.22.0
[tunnelRoutesEnabled] **	*boolean*	OSPF routes over GRE tunnel enabled. **Since: 4.23.1
[vrf]	*string*	VRF name.

2.642.1.1.1 OspfSecondaryDomainId

domainId	*string*	Secondary domain identifier.

2.642.1.1.2 OspfShowInstAreaModel

[areaFilterPrefixList] **	*string*	Area Filter prefix-list. **Since: 4.23.0
[areaFiltersConfigured] **	*boolean*	Area Summary Filters configured. **Since: 4.23.0
areaId	*IPv4 address*	OSPF area ID.
[authType]	*enum: Simple | SHA512 | SHA384 | SHA1 | SHA256 | MD5*	Area authentication type.
dcClearCount **	*integer*	DC clear LSA count. **Since: 4.26.2
indLsaCount **	*integer*	Indication LSA count. **Since: 4.26.2
lsaInformation	*OspfShowInstAreaLsaInfoModel* see 2.642.1.1.2.1	Area LSA information.
normalArea	*boolean*	This area is a normal area.
[nssaInformation]	*OspfShowInstAreaNssaModel* see 2.642.1.1.2.2	Area NSSA information.
numIntf	*integer*	Number of interfaces in the area.

opaqueAreaLsaInformation **	*OspfShowInstAreaScope-OpaqueLsaInfoModel* see 2.642.1.1.2.5	Opaque area scope LSA information. **Since: 4.22.0
opaqueLsaInformation	*OspfShowInstAreaOpaque-LsaInfoModel* see 2.642.1.1.2.3	Area link opaque scope LSA information.
rangeList	*table(IPv4 address) = Ospf-ShowInstAreaRangeModel* see 2.642.1.1.2.4	Dictionary of address ranges in this OSPF area keyed by the area range address.
spfCount	*integer*	Area SPF count.
stubArea	*boolean*	This area is a stub area.
teEnabled **	*boolean*	Traffic engineering enabled in the area. **Since: 4.22.0

2.642.1.1.2.1 OspfShowInstAreaLsaInfoModel

asExternalCksum	*integer*	Area AS external LSA checksum sum.
treeSize	*integer*	Area LSA tree size (number of LSA).

2.642.1.1.2.2 OspfShowInstAreaNssaModel

nssaTransAlways	*boolean*	Nssa translate always flag for area.
[nssaTransInt]	*integer*	Nssa translate interval.
[nssaTransState]	*enum: notTranslator | stopTranslator | translator | extendedTranslator*	Nssa translate state flag for area.

2.642.1.1.2.3 OspfShowInstAreaOpaqueLsaInfoModel

opaqueCksum	*integer*	Area opaque link LSA checksum sum.
opaqueTreeSize	*integer*	Area opaque link tree size (number of opaque link LSA).

2.642.1.1.2.4 OspfShowInstAreaRangeModel

address	*IPv4 address*	Area range address.
cost	*integer*	Area range cost.
[doNotAdvertise] **	*boolean*	Do not advertise the area range. The cost value is invalid if this flag is set to true. **Since: 4.17.0
maskLen	*integer*	Area range mask length.

2.642.1.1.2.5 OspfShowInstAreaScopeOpaqueLsaInfoModel

opaqueCksumSum	*integer*	Area opaque scope LSA checksum sum.
opaqueTreeSize	*integer*	Area opaque scope LSA tree size (number of opaque LSA).

2.642.1.1.3 OspfShowInstExternalLsaInfoModel

asExternalCksum	*integer*	AS external LSA checksum.
treeSize	*integer*	OSPF instance external LSA tree size (number of external LSAs.

2.642.1.1.4 OspfShowInstGracefulRestartInfoModel

gracePeriod	*floating point number*	Maximum time for graceful-restart to complete (secs).
gracefulRestart	*boolean*	Graceful restart mode enabled.
helperLooseLsaCheck	*boolean*	Graceful restart helper allows topology changes.
helperMode	*boolean*	Graceful restart helper mode.
[lastExitReason]	*enum: lastExitHelperAborted* | *lastExitNone* | *lastExitSuccess* | *lastExitNeighborStateChanged* | *lastExitDisabled* | *lastExitGracePeriodExpired*	Reason for last graceful restart exit.
lastExitTime	*floating point number*	Time last graceful restart exited.
[lastRestartReason] **	*enum: Unknown* | *RIB restart* | *ASU2* | *SSO*	Specific reason for last graceful restart. **Since: 4.18.0
plannedOnly	*boolean*	Restrict graceful-restarts to planned events.
restartDuration **	*integer*	Graceful restart duration in microseconds. **Since: 4.18.0
restartExpirationTime	*floating point number*	Time current graceful restart expires.
[state]	*enum: signaled* | *pending* | *exiting* | *none* | *graceful*	Graceful restart speaker state.

2.642.1.1.5 OspfShowInstLsaInfoModel

lsaArrivalInterval	*integer*	LSA arrival interval.
lsaHoldInterval	*integer*	LSA hold interval.
lsaMaxWaitInterval	*integer*	LSA max wait interval.
lsaStartInterval	*integer*	LSA start interval.
numLsa	*integer*	Number of LSAs.

2.642.1.1.6 OspfShowInstMaxLsaInfoModel

maxLsa	*integer*	Maximum number of LSAs allowed.
maxLsaAllowedIgnoreCount	*integer*	Max LSA allowed ignore count.
maxLsaCurrentIgnoreCount	*integer*	Max LSA current ignore count.
maxLsaIgnoreTime	*integer*	Max LSA ignore time (mins).
[maxLsaIgnoreTimeEnd]	*floating point number*	Max LSA ignore time end.
maxLsaIgnoring	*boolean*	Ignoring max LSA condition.
maxLsaResetTime	*integer*	Max LSA reset time (mins).
maxLsaThreshold	*integer*	Max LSA threshold for warning message.

maxLsaWarningOnly	*boolean*	Max LSA warning-only option.

2.642.1.1.7 OspfShowInstMaxMetricTypeInfoModel

maxMetricActive **	*boolean*	Max metric advertisements being currently active or not. **Since: 4.20.0
[maxMetricOnStartUpTime]	*integer*	Time interval on startup when maximum metric is advertised (secs).
[maxMetricTimerLeft]	*integer*	Max metric timer left (secs).
maxMetricType	*integer*	Max metric type.
[maxMetricWaitBgp]	*integer*	Max metric while Bgp is converging.

2.642.1.1.8 OspfShowInstMaxMetricUnsetInfoModel

[maxMetricUnsetOrig-Duration]	*integer*	Max metric originate duration (secs).
[maxMetricUnsetReason]	*enum: maxMetricUnsetUserConfig* | *maxMetricUnsetTimer-Expired* | *maxMetricUnsetBgp-Converged* | *maxMetricUnsetBgpNot-Configured* | *maxMetricUnsetBgpTimer-Expired*	Max metric unset reason.
maxMetricUnsetTstamp	*integer*	Max metric unset timestamp.

2.642.1.1.9 OspfShowInstOpaqueLsaInfoModel

opaque	*boolean*	Opaque LSAs supported.
opaqueCksum	*integer*	Opaque AS LSA checksum.
opaqueTreeSize	*integer*	OSPF instance opaque AS LSA tree size (number of opaque AS LSAs.

2.642.1.1.10 OspfShowInstSpfInfoModel

lastSpf	*integer*	Last SPF run (secs).
nextSpf	*integer*	Next SPF run (msecs).
spfCurrHoldInterval	*integer*	SPF current hold interval.
spfHoldInterval	*integer*	SPF hold interval.
spfInterval	*integer*	SPF interval.
spfMaxWaitInterval	*integer*	SPF max wait interval.
spfStartInterval	*integer*	SPF start interval.

2.643 show ip ospf access-list [...]

see 2.53

2.644 show ip ospf border-routers [...]

vrfs	*table(string) = InstListOspfBorderRoutersInstance* see 2.644.1	OSPF border routers information for all VRFs.

2.644.1 InstListOspfBorderRoutersInstance

instList	*table(numeric string) = OspfBorderRoutersInstance* see 2.644.1.1	OSPF Instance list keyed by instance id.

2.644.1.1 OspfBorderRoutersInstance

ospfBorderRouterEntries	*table(IPv4 address) = OspfBorderRoutersEntry* see 2.644.1.1.1	Dictionary of OSPF border router entries indexed by area-id.
routerId	*IPv4 address*	OSPF router identifier.
shutDown **	*boolean*	Indicates if the OSPF instance is currently shut down. **Since: 4.22.1
vrf	*string*	VRF name.

2.644.1.1.1 OspfBorderRoutersEntry

borderRouters	*table(IPv4 address) = OspfBorderRoutersType* see 2.644.1.1.1.1	Dictionary of OSPF border routers type indexed by router-id.

2.644.1.1.1.1 OspfBorderRoutersType

abr	*boolean*	Router is an area border router.
asbr	*boolean*	Router is an autonomous system boundary router.
metric **	*integer*	Cost to reach this router. **Since: 4.27.2

2.645 show ip ospf counters [...]

vrfs	*table(string) = InstListOspfCountersInstance* see 2.645.1	OSPF Counters.

2.645.1 InstListOspfCountersInstance

instList	*table(numeric string) = OspfCountersInstance* see 2.645.1.1	OSPF Instance list keyed by instance id.

2.645.1.1 OspfCountersInstance

[lastResetTime]	*floating point number*	UTC timestamp of last counters reset.
lsaTypeCounters	*table(string) = OspfCountersLsaEntry* see 2.645.1.1.1	Dictionary of LSA type counters keyed by counter name.
opaqueLsaReceived	*integer*	Number of opaque LSAs received.
selfOriginatedLsaReceived	*integer*	Number of self-originated LSAs received.
selfOriginatedLsaReceived-Deleted	*integer*	Number of self-originated LSAs received and deleted.
selfOriginatedLsaReceived-Updated	*integer*	Number of self-originated LSAs received and updated.
spfCount	*integer*	Number of SPF runs.

2.645.1.1.1 OspfCountersLsaEntry

numAsex	*integer*	Number of AS-external LSAs.
numNetwork	*integer*	Number of network LSAs.
numNssa	*integer*	Number of type-7 NSSA LSAs.
numOpaqueArea	*integer*	Number of opaque area LSAs.
numOpaqueAs	*integer*	Number of opaque AS LSAs.
numOpaqueLink	*integer*	Number of opaque link LSAs.
numRouter	*integer*	Number of router LSAs.
numSummary	*integer*	Number of summary LSAs.
numSummaryAsbr	*integer*	Number of summary ASBR LSAs.

2.646 show ip ospf counters interface [...]

vrfs	*table(string) = InstListOspfCountersInterface* see 2.646.1	OSPF Interface Counters.

2.646.1 InstListOspfCountersInterface

instList	*table(numeric string) = OspfCountersInterface* see 2.646.1.1	OSPF Instance list keyed by instance id.

2.646.1.1 OspfCountersInterface

interfaces	*table(string) = OspfCountersInterfaceEntry* see 2.646.1.1.1	Dictionary of OSPF interfaces indexed by the interface name.

2.646.1.1.1 OspfCountersInterfaceEntry

delayedLsAcksSent	*integer*	Number of delayed LS acknowledgements sent.
[lastResetTime]	*floating point number*	UTC timestamp of the last time counters were reset.
lsAcksReceived	*integer*	Number of LS acknowledgements received.
lsUpdatesInFloodQueue	*integer*	Number of LS updates currently in the flood queue.
lsaType **	*OspfCountersLsaTypeEntry* see 2.646.1.1.1.1	LSA types. **Since: 4.27.0
multicastLsUpdatesSent	*integer*	Number of multicast LS updates sent.

2.646.1.1.1.1 OspfCountersLsaTypeEntry

asExternal	*OspfLsaCounters* see 2.646.1.1.1.2	Number of AS-external LSAs.
network	*OspfLsaCounters* see 2.646.1.1.1.2	Number of network LSAs.
nssa	*OspfLsaCounters* see 2.646.1.1.1.2	Number of type-7 NSSA LSAs.
opaqueArea	*OspfLsaCounters* see 2.646.1.1.1.2	Number of opaque area LSAs.
opaqueAs	*OspfLsaCounters* see 2.646.1.1.1.2	Number of opaque AS LSAs.
opaqueLink	*OspfLsaCounters* see 2.646.1.1.1.2	Number of opaque link LSAs.
router	*OspfLsaCounters* see 2.646.1.1.1.2	Number of router LSAs.
summary	*OspfLsaCounters* see 2.646.1.1.1.2	Number of summary LSAs.
summaryAsbr	*OspfLsaCounters* see 2.646.1.1.1.2	Number of summary ASBR LSAs.

2.646.1.1.1.2 OspfLsaCounters

`received`	*integer*	Number of LSAs received.
`sent`	*integer*	Number of LSAs sent.

2.647 show ip ospf counters interface rates [...]

vrfs	*table(string) = InstListOspfCountersInterfaceRates* see 2.647.1	OSPF Interface Counters Rates.

2.647.1 InstListOspfCountersInterfaceRates

instList	*table(numeric string) = OspfCountersInterfaceRates* see 2.647.1.1	OSPF Instance list keyed by instance id.

2.647.1.1 OspfCountersInterfaceRates

interfaces	*table(string) = OspfCountersInterfaceRatesEntry* see 2.647.1.1.1	Dictionary of OSPF interfaces indexed by the interface name.

2.647.1.1.1 OspfCountersInterfaceRatesEntry

[lastResetTime]	*floating point number*	UTC timestamp of the last time counters were reset.
lsaTypeRates	*OspfCountersLsaRateTypeEntry* see 2.647.1.1.1.1	LSA type rates.

2.647.1.1.1.1 OspfCountersLsaRateTypeEntry

asExternal	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of AS-external LSAs.
network	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of network LSAs.
nssa	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of type-7 NSSA LSAs.
opaqueArea	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of opaque area LSAs.
opaqueAs	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of opaque AS LSAs.
opaqueLink	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of opaque link LSAs.
router	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of router LSAs.
summary	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of summary LSAs.
summaryAsbr	*OspfLsaCountersRate* see 2.647.1.1.1.2	Number of summary ASBR LSAs.

2.647.1.1.1.2 OspfLsaCountersRate

received	*floating point number*	Number of LSAs received per second.

`sent`	*floating point number*	Number of LSAs sent per second.

2.648 show ip ospf counters neighbor [...]

vrfs	*table(string) = InstListOspfCountersNeighbor* see 2.648.1	OSPF Neighbor Counters.

2.648.1 InstListOspfCountersNeighbor

instList	*table(numeric string) = OspfCountersNeighbor* see 2.648.1.1	OSPF Instance list keyed by instance id.

2.648.1.1 OspfCountersNeighbor

neighbors	*list of OspfCountersNeighborEntry* see 2.648.1.1.1	List of neighbor-level counter entries.

2.648.1.1.1 OspfCountersNeighborEntry

databaseDescriptionReceived	*integer*	Number of database description packets received.
databaseDescriptionSent	*integer*	Number of database description packets sent.
interfaceAddress	*IPv4 address*	OSPF neighbor interface address.
interfaceName	*string*	OSPF neighbor interface name.
[lastResetTime]	*floating point number*	UTC timestamp of the last time counters were reset.
lsAcknowledgementSent	*integer*	Number of LS acknowledgements sent.
lsRequestReceived	*integer*	Number of LS requests receieved.
lsRequestSent	*integer*	Number of LS requests sent.
multicastLsAcknowledgementReceived	*integer*	Number of multicast LS acknowledgements received.
multicastLsUpdateReceived	*integer*	Number of multicast LS updates received.
neighborAddress	*IPv4 address*	OSPF neighbor address.
neighborRouterId	*IPv4 address*	OSPF neighbor router identifier.
unicastLsAcknowledgementReceived	*integer*	Number of unicast LS acknowledgements received.
unicastLsUpdateReceived	*integer*	Number of unicast LS updates received.

2.649 show ip ospf database [...]

vrfs	*table(string) = InstListOspfDatabaseInstance* see 2.649.1	OSPF database information.

2.649.1 InstListOspfDatabaseInstance

instList	*table(numeric string) = OspfDatabaseInstance* see 2.649.1.1	OSPF Instance list keyed by instance id.

2.649.1.1 OspfDatabaseInstance

areas	*table(IPv4 address) = OspfAreaModel* see 2.649.1.1.1	A mapping between an area and its LSA list.
externalLsdb	*list of OspfAsExternalLsa* see 2.649.1.1.2	List of LSAs with AS scope LSAs.
type11Lsdb	*list of OspfOpaqueExternalLsa* see 2.649.1.1.3	List of Opaque LSAs with AS scope.

2.649.1.1.1 OspfAreaModel

areaDatabase	*list of OspfAreaScopeLsa* see 2.649.1.1.1.1	Area scoped LSAs.
[interfaces]	*table(string) = OspfInterfaceScopeModel* see 2.649.1.1.1.15	A mapping between an Interface and its LSA list.

2.649.1.1.1.1 OspfAreaScopeLsa

areaLsas	*list of OspfLsa* see 2.649.1.1.1.2	List of LSAs for the area.

2.649.1.1.1.2 OspfLsa

advertisingRouter	*IPv4 address*	LSA advertising router.
age	*integer*	LSA age in seconds.
checksum	*integer*	LSA checksum.
doNotAge	*boolean*	LSA will not be aged.
[exceptionFlag]	*enum: announcingMaximumMetric | announcingMaximumCost*	Indicates maximum cost being advertised for self originated LSAs.
length	*integer*	LSA length.
linkStateId	*IPv4 address*	LSA link state id.

lsaType	*enum: routerLsa* | *asExternalLsa* | *summaryAsbrLsa* | *opaqueType11Lsa* | *summaryLsa* | *opaqueType9Lsa* | *opaqueType10Lsa* | *networkLsa* | *nssaLsa*	LSA type.
options	*OspfDatabaseOptions* see 2.649.1.1.1.3	OSPF LSA options.
[ospfExternalLsa]	*OspfExternalLsa* see 2.649.1.1.1.4	AS or NSSA external LSA.
[ospfNetworkLsa]	*OspfNetworkLsa* see 2.649.1.1.1.5	Network LSA.
[ospfRouterLsa]	*OspfRouterLsa* see 2.649.1.1.1.6	Router LSA.
[ospfSummaryLsa]	*OspfSummaryLsa* see 2.649.1.1.1.9	Summary LSA or ASBR summary LSA.
[ospfTeInfo]	*OspfTrafficEngineeringInfo* see 2.649.1.1.1.10	Traffic Engineering TLV data.
sequenceNumber	*integer*	LSA sequence number.

2.649.1.1.1.3 OspfDatabaseOptions

demandCircuitsSupport	*boolean*	Originating router supports OSPF over Demand Circuits.
doNotUseInRouteCalc	*boolean*	A type 3,5 or 7 LSA is sent from a PE router to a CE router.
externalRoutingCapability	*boolean*	Originating router capable of accepting AS External LSAs.
linkLocalSignaling	*boolean*	OSPF packet contains Link-Local Signaling data block.
multicastCapability	*boolean*	Originating router supports multicast extensions to OSPF.
multitopologyCapability	*boolean*	Originating router supports Multitopology OSPF.
nssaCapability	*boolean*	Originating router supports Type-7 NSSA-External-LSAs.
opaqueLsaSupport	*boolean*	Originating router supports Type 9,10 and 11 Opaque LSAs.

2.649.1.1.1.4 OspfExternalLsa

externalRouteTag	*integer*	External route tag.
forwardingAddress	*IPv4 address*	Forwarding address.
metric	*integer*	Route cost.
metricType	*enum: type1* | *type2*	External metric type (E bit) type1 : External metric Type 1 type2 : External metric Type 2.
networkMask	*IPv4 address*	IP address mask for the advertised destination.

2.649.1.1.1.5 OspfNetworkLsa

attachedRouters	*list of IPv4 address*	List of routers attached to the network.
networkMask	*IPv4 address*	IP address mask for the network.

2.649.1.1.1.6 OspfRouterLsa

numRtrLinks	*integer*	Number of router links.
routerLsaLinks	*list of OspfRouterLsaLinkEntry* see 2.649.1.1.1.7	Links for this router.
rtrLsaOptions	*OspfRouterLsaOptions* see 2.649.1.1.1.8	Router lsa options.

2.649.1.1.1.7 OspfRouterLsaLinkEntry

linkData	*IPv4 address*	Router link data.
linkId	*IPv4 address*	Router link ID.
linkType	*enum: transitNetwork | unknown | virtualLink | stubNetwork | pointToPoint*	Description of the router link.
metric	*integer*	Cost of the router link.
numTos	*integer*	Number of TOS metrics for the link.

2.649.1.1.1.8 OspfRouterLsaOptions

areaBorderRouter	*boolean*	This is an area border router.
asBoundaryRouter	*boolean*	This is an autonomous system boundary router.
nssaTranslation	*boolean*	Translate nssa lsas to as external lsas.
virtualLinkEndpoint	*boolean*	This is an endpoint for virtual links in the area.

2.649.1.1.1.9 OspfSummaryLsa

metric	*integer*	Route cost.
networkMask	*IPv4 address*	IP address mask for the destination.

2.649.1.1.1.10 OspfTrafficEngineeringInfo

[linkTlvInfo]	*LinkAddressTlv* see 2.649.1.1.1.11	Link TLV.
malformedTlv	*boolean*	The TLV is not parsable.
[rtrTlvInfo]	*RtrAddressTlv* see 2.649.1.1.1.13	Router Address TLV.
[tlvLength]	*integer*	Top Level TLV length.
[tlvName]	*enum: link | routerAddress | unsupported*	Top Level TLV Name.
[tlvType]	*integer*	Top Level TLV type.

[unsupportedSubTlvInfo]	*list of UnsupportedSubTlv* see 2.649.1.1.1.14	Unsupported sub-TLV information.

2.649.1.1.1.11 LinkAddressTlv

[adminGroup]	*integer*	Administrative group assigned to the link.
linkId	*IPv4 address*	Ip Address of the other end of the link.
linkType	*enum: pointToPoint | unknown | multiAccess*	Description of the link.
[localIpAddresses]	*list of IPv4 address*	IP addresses of the link.
[maxLinkBw]	*integer*	Maximum bandwidth (bps) that can be used on the link.
[maxReservableBw]	*integer*	Maximum bandwidth (bps) that can be reserved on the Link.
[maxReservablePriorityBw]	*TeReservablePriorityBandwidth* see 2.649.1.1.1.12	Maximum bandwidth reservable for a priority.
[remoteIpAddresses]	*list of IPv4 address*	IP addresses of the neighbors of the link.
[srlgs]	*list of integer*	Shared risk link group identifiers.
[teMetric]	*integer*	The cost of the link.

2.649.1.1.1.12 TeReservablePriorityBandwidth

priority0	*integer*	Reservable bandwidth (bps) for priority level 0.
priority1	*integer*	Reservable bandwidth (bps) for priority level 1.
priority2	*integer*	Reservable bandwidth (bps) for priority level 2.
priority3	*integer*	Reservable bandwidth (bps) for priority level 3.
priority4	*integer*	Reservable bandwidth (bps) for priority level 4.
priority5	*integer*	Reservable bandwidth (bps) for priority level 5.
priority6	*integer*	Reservable bandwidth (bps) for priority level 6.
priority7	*integer*	Reservable bandwidth (bps) for priority level 7.

2.649.1.1.1.13 RtrAddressTlv

teRouterId	*IPv4 address*	Traffic Engineering IPv4 Router-Id.

2.649.1.1.1.14 UnsupportedSubTlv

tlvLength	*integer*	TLV length.
tlvType	*integer*	TLV type.

2.649.1.1.1.15 OspfInterfaceScopeModel

linkLsas	*list of OspfType9LsaList* see 2.649.1.1.1.16	List of LSAs for the area.

2.649.1.1.1.16 OspfType9LsaList

ospfType9Lsas	*list of OspfLsa* see 2.649.1.1.1.2	List of LSAs for the interface.

2.649.1.1.2 OspfAsExternalLsa

externalLsas	*list of OspfLsa* see 2.649.1.1.1.2	List of AS external LSAs.

2.649.1.1.3 OspfOpaqueExternalLsa

type11Lsas	*list of OspfLsa* see 2.649.1.1.1.2	Type 11 Opaque LSAs.

2.650 show ip ospf database database-summary [...]

vrfs	*table(string) = InstListOspf-DatabaseSummary* see 2.650.1	OSPF database summary information.

2.650.1 InstListOspfDatabaseSummary

instList	*table(numeric string) = OspfDatabaseSummary* see 2.650.1.1	OSPF Instance list keyed by instance id.

2.650.1.1 OspfDatabaseSummary

numAsex	*integer*	Number of AS-external LSAs.
numNetwork	*integer*	Number of network LSAs.
numNssa	*integer*	Number of type-7 NSSA LSAs.
numOpaqueArea	*integer*	Number of opaque area LSAs.
numOpaqueAs	*integer*	Number of opaque AS LSAs.
numOpaqueLink	*integer*	Number of opaque link LSAs.
numRouter	*integer*	Number of router LSAs.
numSummary	*integer*	Number of summary LSAs.
numSummaryAsbr	*integer*	Number of summary ASBR LSAs.

2.651 show ip ospf interface [...]

vrfs	*table(string) = InstListOspfInterface* see 2.651.1	OSPF instance information for all VRFs.

2.651.1 InstListOspfInterface

instList	*table(numeric string) = OspfInterface* see 2.651.1.1	OSPF Instance list keyed by instance id.

2.651.1.1 OspfInterface

interfaces	*table(string) = OspfInterfaceEntry* see 2.651.1.1.1	Dictionary of OSPF Interfaces indexed by the interface name.

2.651.1.1.1 OspfInterfaceEntry

[activeMetricRule] **	*string*	Active Metric Profile Rule. **Since: 4.29.2
area	*IPv4 address*	Area assigned to the interface.
[authKeyId] **	*integer*	Authentication Key ID. **Since: 4.22.0
[authentication]	*enum: simple | sha256 | sha512 | sha384 | sha1 | md5*	Authentication algorithm.
[backupDesignatedRouter]	*IPv4 address*	Backup designated router ID.
bfdState	*enum: disabled | default | enabled*	BFD configuration state.
cost	*integer*	Interface metric.
deadInterval	*integer*	Dead interval in seconds.
[designatedRouter]	*IPv4 address*	Designated router ID.
helloInterval	*integer*	Hello interval in seconds.
interfaceAddress	*IPv4 address*	Interface IP address.
interfaceMask	*integer*	Interface IP mask length.
[interfaceSpeed] **	*integer*	Interface Speed in mbps. **Since: 4.29.2
interfaceType	*enum: p2p | nbma | p2mp | virtual | broadcast*	Network interface type.
[metricProfile] **	*string*	Metric Profile Name. **Since: 4.29.2
[mplsLdpIgpSync]	*boolean*	MPLS LDP-IGP Synchronization, ensures that LDP is fully established before IGP path is used for switching. Uses global setting as default.
neighborCount	*integer*	Neighbour count.
options	*OspfOptions* see 2.651.1.1.1.1	Interface optional capabilities.
passive	*boolean*	Passive state.
[priority]	*integer*	Interface priority.

[remoteAddress] **	*IPv4 address*	Interface remote IP address. **Since: 4.30.1
retransmitInterval	*integer*	Retransmit interval in seconds.
state	*enum: p2p | loopback | backupDr | drOther | down | dr | waiting*	Interface state.
teEnabled **	*boolean*	Traffic engineering state. **Since: 4.22.0
transmitDelay	*integer*	Transmit delay in seconds.

2.651.1.1.1.1 OspfOptions

Could also be an instance of:
OspfDatabaseOptions 2.649.1.1.1.3

demandCircuitsSupport	*boolean*	Originating router supports OSPF over Demand Circuits.
doNotUseInRouteCalc	*boolean*	A type 3,5 or 7 LSA is sent from a PE router to a CE router.
externalRoutingCapability	*boolean*	Originating router capable of accepting AS External LSAs.
linkLocalSignaling	*boolean*	OSPF packet contains Link-Local Signaling data block.
multicastCapability	*boolean*	Originating router supports multicast extensions to OSPF.
multitopologyCapability	*boolean*	Originating router supports Multitopology OSPF.
nssaCapability	*boolean*	Originating router supports Type-7 NSSA-External-LSAs.
opaqueLsaSupport	*boolean*	Originating router supports Type 9,10 and 11 Opaque LSAs.

2.652 show ip ospf lsa-log [...]

vrfs	*table(string) = InstListOspfLsaLogModel* see 2.652.1	OSPF lsa log information.

2.652.1 InstListOspfLsaLogModel

instList	*table(numeric string) = OspfLsaLogModel* see 2.652.1.1	OSPF Instance list keyed by instance id.

2.652.1.1 OspfLsaLogModel

lsaLogEntries	*list of OspfLsaLogEntry* see 2.652.1.1.1	List of OSPF LSA log entries.
routerId	*IPv4 address*	Router ID.
vrf	*string*	VRF name.

2.652.1.1.1 OspfLsaLogEntry

advertisingRouter	*IPv4 address*	LSA advertising router.
linkStateId	*IPv4 address*	LSA link state identifier.
logTime	*floating point number*	LSA log generation time.
lsaEventType	*enum: outDelay | backoffRestarted | discarded | backoffStarted | maxWaitChanged | backedOff*	LSA event types: backoffStarted - lsa backoff started backedOff - lsa backed off backoffRestarted - lsa backoff restarted outDelay - lsa delayed discarded - lsa discarded maxWaitChanged - lsa max wait value changed.
lsaHoldInterval	*integer*	LSA Hold interval in milliseconds.
lsaType	*integer*	LSA type.
routerNetworkMask	*integer*	LSA advertising router mask.
sequenceNumber	*integer*	LSA sequence number.

2.653 show ip ospf mpls ldp sync [...]

vrfs	*table(string) = InstListOspfMplsLdpSyncModel* see 2.653.1	OSPF mpls ldp sync information for all VRFS.

2.653.1 InstListOspfMplsLdpSyncModel

instList	*table(numeric string) = OspfMplsLdpSyncModel* see 2.653.1.1	OSPF Instance list keyed by instance id.

2.653.1.1 OspfMplsLdpSyncModel

interfaces	*table(string) = OspfMplsLdpSyncEntry* see 2.653.1.1.1	OSPF MPLS LDP Sync status.

2.653.1.1.1 OspfMplsLdpSyncEntry

[holddownTime]	*integer*	IGP holddown time.
[igpDelay]	*integer*	IGP link notify delay.
instanceId	*integer*	OSPF Instance ID.
interface	*string*	OSPF interface.
ldpEnabled	*boolean*	Ldp enabled.
syncAchieved	*enum: No | Not applicable | Yes*	Sync achieved.
syncRequired	*boolean*	Sync required.

2.654 show ip ospf neighbor [...]

vrfs	*table(string) = InstListOspfNeighborsInstance* see 2.654.1	OSPF neighbor instance information for all VRFs.

2.654.1 InstListOspfNeighborsInstance

instList	*table(numeric string) = OspfNeighborsInstance* see 2.654.1.1	OSPF Instance list keyed by instance id.

2.654.1.1 OspfNeighborsInstance

ospfNeighborEntries	*list of OspfNeighborsEntry* see 2.654.1.1.1	List of OSPF neighbor entries.

2.654.1.1.1 OspfNeighborsEntry

adjacencyState	*enum: attempt | gracefulRestart | exchStart | 2Ways | loading | down | init | full | exchange*	OSPF neighbor adjacency state. down - No hello received from neighbor attempt - Router attempting to establish connection init - Hello received but two-way communication not established 2Ways - Bi-directional communication established between two routers exchStart - Exchange of link state information between routers and their DR and BDR started exchange - Routers exchanging database descriptor packets loading - Exchange of link state information occuring full - Routers are fully adjacent with each other gracefulRestart - Neighbor is gracefully restarting.
[details]	*OspfNeighborDetails* see 2.654.1.1.1.1	OSPF neighbor details.
drState	*enum: DROTHER | DR | BDR*	OSPF neighbor DR state. DR - Neighbor is the designated router BDR - Neighbor is the backup designated router DROTHER - Neighbor is neither DR or the BDR.
inactivity	*floating point number*	Timestamp of OSPF neighbor inactivity/dead timer.
interfaceAddress	*IPv4 address*	OSPF neighbor interface address.
interfaceName	*string*	OSPF neighbor interface name.
options	*OspfOptions* see 2.651.1.1.1.1	OSPF neighbor options.
priority	*integer*	OSPF neighbor priority.
routerId	*string*	OSPF neighbor router identifier.
[routerName]	*string*	OSPF neighbor router domain name.

2.654.1.1.1.1 OspfNeighborDetails

areaId	*IPv4 address*	OSPF neighbor area identifier.

backupDesignatedRouter	*IPv4 address*	OSPF neighbor backup designated router id.
bfdRequestSent	*boolean*	OSPF neighbor BFD request is sent.
bfdState	*enum: down | adminDown | up | init*	OSPF neighbor BFD state. adminDown - Session taken down for administrative purposes init - Waiting for session establishment up - Session established successfully down - Session is down or has just been created.
designatedRouter	*IPv4 address*	OSPF neighbor designated router identifier.
[grExitReason] **	*enum: grExitGracePeriodExpired | grExitNone | grExitNetworkChanged | grExitHelperDisabled | grExitAdjNotFull | grExitGraceLsaFlushed | grExitGracefulRestartInProgress*	OSPF neighbor graceful restart exit reason. **Since: 4.15.3
grHelperTimer **	*floating point number*	Timestamp of OSPF neighbor graceful period expiration. **Since: 4.15.3
grLastRestartTime **	*floating point number*	Timestamp of the OSPF neighbor last graceful restart time. **Since: 4.15.3
grNumAttempts **	*integer*	OSPF neighbor graceful restart attempts. **Since: 4.15.3
inactivityDefers	*integer*	OSPF neighbor inactivity deferred counter.
numberOfStateChanges	*integer*	OSPF neighbor number of state changes.
retransmissionCount	*integer*	OSPF neighbor retransmission count.
[srNeighborInfo] **	*OspfSegmentRoutingNeighborInfo* see 2.654.1.1.1.2	OSPF SR neighbor details. **Since: 4.23.1
stateTime	*floating point number*	OSPF neighbor current state time.

2.654.1.1.1.2 OspfSegmentRoutingNeighborInfo

srGbBase	*integer*	SR Global Block base.
srGbRange	*integer*	SR Global Block range.

2.655 show ip ospf neighbor adjacency-changes [...]

vrfs	*table(string) = InstListOspfNeighborAdjChanges* see 2.655.1	OSPF adjacency changes info for all VRFs.

2.655.1 InstListOspfNeighborAdjChanges

instList	*table(numeric string) = OspfNeighborAdjChanges* see 2.655.1.1	OSPF Instance list keyed by instance id.

2.655.1.1 OspfNeighborAdjChanges

adjacencyChanges	*list of OspfNeighborAdjChangesEntry* see 2.655.1.1.1	List of OSPF adjacency changes.
routerId **	*IPv4 address*	OSPF router identifier. **Since: 4.22.0
shutDown **	*boolean*	Indicates if the OSPF instance is currently shut down. **Since: 4.22.1

2.655.1.1.1 OspfNeighborAdjChangesEntry

adjChangeTimeStamp	*floating point number*	Time when interface adjacency changes in UTC.
adjEstablished	*boolean*	OSPF neighbor interface is established.
[dropReason]	*string*	Reason for the dropped interface adjacency.
interfaceName	*string*	OSPF interface where adjacency is established.
neighborAddress	*IPv4 address*	OSPF neighbor address.
neighborRouterId	*IPv4 address*	OSPF neighbor router identifier.
[routerName]	*string*	OSPF neighbor router domain name.

2.656 show ip ospf neighbor summary [...]

vrfs	*table(string) = InstListOspfNeighborSummaryInstance* see 2.656.1	OSPF neighbor summary for all VRFs.

2.656.1 InstListOspfNeighborSummaryInstance

instList	*table(numeric string) = OspfNeighborSummaryInstance* see 2.656.1.1	OSPF Instance list keyed by instance id.

2.656.1.1 OspfNeighborSummaryInstance

neighborSummary	*OspfNeighborSummary* see 2.656.1.1.1	OSPF neighbor summary information.

2.656.1.1.1 OspfNeighborSummary

num2Ways	*integer*	Number of neighbors in 2 Ways state.

numAttempt	*integer*	Number of neighbors in Attempt state.
numDown	*integer*	Number of neighbors in Down state.
numExchStart	*integer*	Number of neighbors in Exchange Start state.
numExchange	*integer*	Number of neighbors in Exchange state.
numFull	*integer*	Number of neighbors in Full state.
numGracefulRestart	*integer*	Number of neighbors in Graceful Restart state.
numInit	*integer*	Number of neighbors in Init state.
numLoading	*integer*	Number of neighbors in Loading state.

2.657 show ip ospf sham-link [...]

see 2.651

2.658 show ip ospf sham-link route [...]

revision 4

[vrfs]	*table(string) = RibRoutesByProtocol* see 2.658.1	For each VRF, map of routes learned through routing protocols that are considered for installation into the forwarding table.

2.658.1 RibRoutesByProtocol

[ribRoutesByProtocol]	*table(string) = RibRoutes* see 2.658.1.1	Map of routes learned through routing protocols that are considered for installation into the forwarding table.

2.658.1.1 RibRoutes

ribRoutes	*table(IPv4 or IPv6 address prefix) = RibRoute* see 2.658.1.1.1	List of routes learned through routing protocolsfor a given prefix.

2.658.1.1.1 RibRoute

bestRoute	*boolean*	Set to true if this route was selected for installation into the forwarding table, false otherwise.
lastUpdateEpochTime	*integer*	Timestamp in UTC time of the last time the route has been updated.
loopedRoute	*boolean*	Set to true if this route is part of a recursive route resolution loop, false otherwise.
metric	*integer*	The routing protocols internal metric (cost) for this route.
metricType	*enum: MED | metric | AIGP*	The type of metric.
preference	*integer*	The administrative distance of the route.
[recmpEligible]	*boolean*	Set to true if this route is marked eligible for RECMP.
[recursiveNextHops]	*list of RecursiveNextHop* see 2.658.1.1.1.1	List of next-hop routers through which this route is reachable.
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	List of immediate next-hop routers through which this route is reachable.
[srcVrfId]	*integer*	ID of VRF from which this route was leaked.
[srcVrfName]	*string*	Name of VRF from which this route was leaked.

2.658.1.1.1.1 RecursiveNextHop

[Weight]	*floating point number*	The weight associated with this next-hop.
[colors]	*list of ColorAndCoBits* see 2.658.1.1.1.2	The list of SR-TE color values and the CO bits for the recursive nexthop.
[flags]	*ViaFlagDetails* see 2.658.1.1.1.8	Via Flags.

[linkBandwidth]	*floating point number*	The link bandwith associated with this next-hop in bytes per second.
[loopedNextHop]	*boolean*	Set to true if this next-hop is part of a recursive route resolution loop, false otherwise.
[metric]	*integer*	The routing protocols internal metric (cost) for the route used to reach this next-hop.
[metricType]	*enum: MED* | *metric* | *AIGP*	The type of metric.
[nextHop]	*IP address*	Next hop router address.
[preference]	*integer*	The administrative distance of the route used to reach this next-hop.
[resolutionRibs]	*ResolutionRibProfile* see 2.658.1.1.1.3	Resolution RIBs.
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	The list of immediate next-hop routers through which this recursive next-hop is reachable.
transportType	*enum: ipv4* | *ipv6* | *evpn* | *mpls*	The type of transport used to reach the destination.
[unresolvedNextHop]	*boolean*	Set to true if this next-hop is unresolved, false otherwise.
[viaSetType]	*enum: ipv4* | *ipv6* | *invalid* | *mpls* | *tunnel* | *nexthop-group* | *evpn*	Type of vias in the via set.
[viaType]	*enum: backup* | *active*	Type of via, either active or backup.

2.658.1.1.1.2 ColorAndCoBits

coBits	*enum: anyMatch* | *nullMatch* | *exactMatch*	The type of SR-TE policy matching to be done for this color.
color	*integer*	Color value associated with the nexthop.

2.658.1.1.1.3 ResolutionRibProfile

[error]	*string*	Invalid resolution profile reason.
resolutionMethods	*list of ResolutionRib* see 2.127.1.3.2.15	List of resolution methods.

2.658.1.1.1.4 ResolvedNextHop

[evpnNextHop]	*EvpnNextHop* see 2.658.1.1.1.5	Vtep used to reach a destination overevpn transport.
[ipNextHop]	*ResolvedIpNextHop* see 2.658.1.1.1.6	Immediate next-hop router used to reach a destination over ipv4/ipv6 transport.
[notInFec] **	*boolean*	Not in FEC. **Since: 4.24.0
[ospfShamLinkNextHop] **	*ShamLinkNextHop* see 2.658.1.1.1.7	Sham-link interface used as next hop to reach the destination. **Since: **currentRelease**
transportType	*enum: ipv4* | *ipv6* | *evpn*	The type of transport used to reach the destination.

[vrfName] **	*string*	Name of VRF in which this next-hop was learned. **Since: 4.23.1

2.658.1.1.1.5 EvpnNextHop

[localInterface] **	*interface name*	Name of the source interface. **Since: 4.26.0
routerMac	*MAC address*	Router mac address of the vtep.
vni	*integer*	Vni associated with this next-hop.
vtep	*IP address*	Vtep address.

2.658.1.1.1.6 ResolvedIpNextHop

[arpResolved]	*boolean*	Designates if nextHop is resolved in ARP/ND.
[decapActionType] **	*enum: none | ipip*	Decapsulation type. **Since: 4.27.0
[fallbackVrf] **	*string*	Fallback VRF name. **Since: 4.27.2
interface	*interface name*	Interface over which the next-hop router is reachable.
[label] **	*integer*	Mpls label. **Since: 4.21.3
[nextHop]	*IP address*	Next hop router address.
[nexthopGroupName] **	*string*	Nexthop Group name over which via resolves. **Since: 4.20.5
[tunnelInterface] **	*TunnelInfo* see 2.151.1.3.1	Tunnel end point details. **Since: 4.21.3
[tunnelVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	LSP's for the tunnel. **Since: 4.21.3

2.658.1.1.1.7 ShamLinkNextHop

interface	*string*	Interface over which the next-hop router is reachable.
[nextHop]	*IP address*	Next hop router address.

2.658.1.1.1.8 ViaFlagDetails

[arpResolved]	*boolean*	ARP/ND resolution information is available for this via.
[bfdMonitoredVia]	*boolean*	The via is monitored by BFD.
[martianVia]	*boolean*	This via is for a martian route.
[mustResolveViaConnected]	*boolean*	Via must recursively resolve only over connected routes.
[requireMplsResolution]	*boolean*	Via may recursively resolve over mpls tunnel or RIB/connected routes.
[routeCacheConnectedVia]	*boolean*	This via is for a route cache connected route.

[suppressResolution]	*boolean*	Resolution status information is not published to protocols.

2.659 show ip ospf spf-log [...]

vrfs	*table(string) = InstListOspfSpfLog* see 2.659.1	OSPF SPF log information.

2.659.1 InstListOspfSpfLog

instList	*table(numeric string) = OspfSpfLog* see 2.659.1.1	OSPF Instance list keyed by instance id.

2.659.1.1 OspfSpfLog

[routerId]	*IPv4 address*	OSPF router identifier.
spflogEntries	*list of OspfSpfLogEntry* see 2.659.1.1.1	OSPF SPF Log entries.

2.659.1.1.1 OspfSpfLogEntry

spfDuration	*floating point number*	SPF run duration in milliseconds.
spfStartTime	*floating point number*	SPF Start time in UTC.

2.660 show ip ospf summary [...]

vrfs	*table(string) = InstListOspfShowSummaryModel* see 2.660.1	OSPF instance information for all VRFs.

2.660.1 InstListOspfShowSummaryModel

instList	*table(numeric string) = OspfShowSummaryModel* see 2.660.1.1	OSPF Instance list keyed by instance id.

2.660.1.1 OspfShowSummaryModel

inst	*_OspfShowInstModelAcc* see 2.660.1.1.2	OSPF instance information.
lsasByArea	*table(string) = OspfDatabaseSummary* see 2.650.1.1	Database info by area.
neighbors	*_OspfNeighborsInstanceAcc* see 2.660.1.1.1	Neighbor information.

2.660.1.1.1 _OspfNeighborsInstanceAcc

ospfNeighborEntries	*list of OspfNeighborsEntry* see 2.654.1.1.1	List of OSPF neighbor entries.

2.660.1.1.2 _OspfShowInstModelAcc

abr	*boolean*	This OSPF instance is an Area Border Router.
[areaList]	*table(IPv4 address) = OspfShowInstAreaModel* see 2.642.1.1.2	List of areas in this OSPF instance.
asbr	*boolean*	This OSPF instance is an AS Border Router.
instanceId	*integer*	OSPF instance ID.
[lsaInformation]	*OspfShowInstLsaInfoModel* see 2.642.1.1.5	LSA information.
maxLsaInformation	*OspfShowInstMaxLsaInfoModel* see 2.642.1.1.6	Max LSA information.
[primaryDomainId] **	*string*	Primary domain identifier. **Since: **currentRelease**
[routerId]	*IPv4 address*	OSPF Router id.
[secondaryDomainIds] **	*list of OspfSecondaryDomainId* see 2.642.1.1.1	List of secondary domain identifiers. **Since: **currentRelease**
shutDown **	*boolean*	Indicates if the OSPF instance is currently shut down. **Since: 4.20.5

[spfInformation]	*OspfShowInstSpfInfoModel* see 2.642.1.1.10	SPF information.
[vrf]	*string*	VRF name.

2.661 show ip pim access-list [...]

see 2.53

2.662 show ip pim anycast-rp [...]

anycastRpAddrs	*list of IP address*	a List of Anycase RP addresses.
peerAddrs	*list of IP address*	a List of Anycase peers addresses.

2.663 show ip pim bidirectional df [...]

rpaDfInfo	*table(Interface name) = Rpas* see 2.663.1	Mapping from interface to DF information.

2.663.1 Rpas

dfInfo	*table(IPv4 or IPv6 address) = DfInfo* see 2.663.1.1	Mapping from RP address to DF information.

2.663.1.1 DfInfo

[df]	*boolean*	If true, the router is the DF for the specified interface.
[dfAddr]	*IP address*	DF IP address.
dfState	*enum: lose | win | backoff | offer*	Current DF election state.
[intf] **	*interface name*	Blah Interface. **Since: 4.22.0
[routeMetric]	*integer*	DF route metric.
[routePreference]	*integer*	DF route preference.

2.664 show ip pim bidirectional rp [...]

see 2.23

2.665 show ip pim bidirectional rp-hash GROUP_ADDR [...]

see 2.24

2.666 show ip pim bidirectional upstream joins [...]

see 2.25

2.667 show ip pim bsr [...]

electedBsrs	*table(IPv4 or IPv6 address prefix) = ElectedBsr* see 2.667.1	Map of Bootstrap Routers per Administrative scoped Zone.

2.667.1 ElectedBsr

eBsrAddress	*IP address*	Elected BSR Address.
eBsrCreationTime	*floating point number*	UTC time at which the BSR was elected.
eBsrExpires	*floating point number*	Time after which this BSR might expire.
eBsrExpiryTime	*floating point number*	UTC time at which the BSR will expire.
eBsrHashMaskLen	*integer*	BSR Hash Mask Length.
eBsrPriority	*integer*	BSR priority.
eBsrUpTime	*floating point number*	Time for which this BSR has been elected.
eBsrZone	*IP address prefix*	Administrative Scope Zone.
[local]	*boolean*	Indicates of this router is the BSR.
[scope]	*string*	Hints if the BSR is global or admin scoped.

2.668 show ip pim bsr counters [...]

[errorCounters]	*PimBsrErrorCounters* see 2.668.1	Error counters.
[rx]	*integer*	Total packets received.
[rxBsm]	*integer*	Total BSM's received.
[rxBsmError]	*integer*	Total BSMs received and dropped due to errors.
[rxBsmSuccess]	*integer*	Total BSMs successfully processed.
[rxCrpAdv]	*integer*	Total CRP Advertisement messages received.
[rxCrpAdvError]	*integer*	Total Crp Advertisements received and dropped due toerrors or configuration.
[rxCrpAdvSuccess]	*integer*	Total CRP Advertisements successfully processed.
[rxError]	*integer*	Total packets received and dropped due to errors.
[rxSuccess]	*integer*	Total packets received and processed successfully.
[tx]	*integer*	Total packets transmitted.
[txError]	*integer*	Total packets transmitted and dropped due to errors.
[txSuccess]	*integer*	Total packets transmitted successfully.

2.668.1 PimBsrErrorCounters

rxFilter	*integer*	Total packets received and dropped due to filters.
rxInvalid	*integer*	Total invalid BSMs received.
rxNbrRpfCheck	*integer*	Total BSMs dropped due to RPF check failure.
rxNoForwardBsm	*integer*	Total BSMs dropped due to no forward flag.
rxScopeMismatch	*integer*	Total BSMs dropped due to scope mismatch.
rxUnicastBsm	*integer*	Total BSMs dropped due to unicast destination.

2.669 show ip pim bsr debug

af	*enum: ipv4 | ipv6 | ipunknown*	Address Family supported in this instance.
bootstrapRouter	*BootstrapRouter* see 2.667	Elected Bootstrap Router.
bsBorder	*table(IPv4 or IPv6 address prefix) = PimBsrBorder* see 2.669.1	Pim BSR Border Map.
bsTimeout	*integer*	Boorstrap Holdtime.
cbsr	*table(IPv4 or IPv6 address prefix) = PimBsrCandidate* see 2.669.2	Candidate BSR Map.
counters	*PimBsrCounters* see 2.668	Pim BSR packet counters.
crp	*table(IPv4 or IPv6 address) = PimBsrRpCandidate* see 2.669.3	Candidate RP Map.
crpPermit	*list of IP address prefix*	Candidate RP permit list.
crpStatus	*table(IPv4 or IPv6 address prefix) = PimBsrRpCandidateStatus* see 2.669.4	C-RP sets for G-RP mapping.
szTimeout	*integer*	Scope Zone timeout interval.
zoneAcl	*table(string) = ZoneAcl* see 2.669.5	IntfId to Acl map.
zoneGroup	*table(IPv4 or IPv6 address prefix) = ZoneGroup* see 2.669.6	IntfId to Zone map.

2.669.1 PimBsrBorder

intfs	*list of Interface name*	Interface list.
prefix	*IP address prefix*	Group Prefix.

2.669.2 PimBsrCandidate

address	*IP address*	Candidate BSR IP.
bsPeriod	*integer*	Bootstrap message interval.
elected	*boolean*	Indicates if the candidate BSR is elected BSR.
hashMaskLen	*integer*	Candidate BSR hash mask length.
intfId	*interface name*	Candidate BSR interface identifier.
priority	*integer*	Candidate BSR priority.
state	*enum: eBsr | cBsr | pBsr | noInfoBsr*	Current state of candidate BSR as per RFC.
zone	*IP address prefix*	Zone.

2.669.3 PimBsrRpCandidate

address	*IP address*	Candidate RP IP addres.
interval	*integer*	Candidate RP advertisement interval.
intfId	*interface name*	Candidate RP interface identifier.
prefixOptions	*table(IPv4 or IPv6 address prefix) = integer*	Group prefixes and associated priorities.
prefixSet	*list of IP address prefix*	Group prefixes served by the candidate RP.

2.669.4 PimBsrRpCandidateStatus

crpSet	*table(IPv4 or IPv6 address prefix) = PimRpCandidateSet* see 2.23.1	C-RP set map.
rpSetType	*enum: noneRp | bsrRp | staticRp*	Type of RP Set.
zone	*IP address prefix*	Zone.

2.669.5 ZoneAcl

intfId	*interface name*	Interface Identifier.
zone	*string*	Administrative Scoped Zone.

2.669.6 ZoneGroup

intfId	*interface name*	Interface Identifier.
zone	*IP address prefix*	Administrative Scoped Zone.

2.670 show ip pim bsr rp [...]

see 2.23

2.671 show ip pim bsr rp-hash GROUP [...]

see 2.24

2.672 show ip pim config-sanity anycast-rp [...]

downIntfs	*list of Interface name*	Anycast RP interfaces which are down.
notAnycastRPs	*list of IP address*	This RP address is not configured as anycast RP.
notCfgRPs	*list of IP address*	Anycast RPs not configured on the router.
notIntfCfgRPs	*list of IP address*	Anycast RP not configured on any interface.
notLoopback	*list of Interface name*	Anycast RP interfaces which are not loopback.

notRPs	*list of IP address*	Anycast RPs not configured as RP addresses.
routeCheck	*list of RouteCheck* see 2.672.1	route check hints.

2.672.1 RouteCheck

hintsFound	*integer*	number of hints found.
noRoute	*boolean*	No route to the prefix.
nodeType	*string*	Node type.
nonPimHop	*list of IPv4 address*	Route via non-Pim hops.
nonPimIntf	*list of Interface name*	Route via non-Pim interface.
prefix	*IPv4 address*	Route check for this prefix.

2.673 show ip pim config-sanity bidirectional [...]

[mroute]	*MrtConfigCheck* see 2.673.1	mroute config check for PIM Bidirectional mode.
[pimBidirEnabled]	*boolean*	PIM Bidirectional enabled.
[pimBidirSupported]	*boolean*	PIM Bidirectional supported platform.
[staticRpSet]	*StaticRpSet* see 2.673.2	staticRp config check for PIM Bidirectional mode.

2.673.1 MrtConfigCheck

hintsFound	*integer*	number of hints found.
noRpGroups	*list of IP address*	Group without RP.
rcModel	*list of RouteCheck* see 2.672.1	Route config check.
uroutes	*integer*	number of unresolved routes.

2.673.2 StaticRpSet

hintsFound	*integer*	number of hints found.
noDefaultRp	*boolean*	No default RP configured.
noRp	*boolean*	No RP configured.
overlapSet	*list of OverlapGroup* see 2.673.2.1	List of overlaping group ranges.
rcModel	*list of RouteCheck* see 2.672.1	Route config check.
rpSet	*list of StaticRp* see 2.673.2.2	List of group RP mappings.

2.673.2.1 OverlapGroup

group	*IP address prefix*	Group address prefix.
overlaps	*list of IP address prefix*	List of overlaping group ranges.

2.673.2.2 StaticRp

mGroup	*string*	group address.
mRpList	*list of string*	List of RPs configured.

2.674 show ip pim config-sanity igmp [...]

boundaryIntfs	*list of Interface name*	Interfaces with multicast boundaries.
disabledIntfs	*list of Interface name*	Disabled interfaces.
hintsFound	*integer*	number of hints found.
invalidBoundaryGroups	*list of InvalidBoundaryGroup* see 2.674.1	Invalid groups configured for boundary.
noRpGroupc	*list of IP address*	Connected group without RP.
noRpGroups	*list of IP address*	Static group without RP.
nonDrIntfs	*list of Interface name*	Non DR interfaces.
nonDrSgIntfs	*list of Interface name*	Interfaces which are not DR.
notDefinedBAcl	*list of NotDefinedBAcl* see 2.674.2	Not defined Acl.
notStandardBAcl	*list of NotStandardBAcl* see 2.674.3	Not standard Acl.
rcModel	*list of RouteCheck* see 2.672.1	Route config check.

2.674.1 InvalidBoundaryGroup

group	*IpAddrAndMask* see 2.241.1.2.2.9	Invalid group prefix.
reason	*string*	Reason for the group being invalid.

2.674.2 NotDefinedBAcl

acl	*string*	Acl name.
intf	*interface name*	Interface Name.

2.674.3 NotStandardBAcl

acl	*string*	Acl name.
intf	*interface name*	Interface Name.

2.675 show ip pim config-sanity interface [...]

disabled	*boolean*	PIM is disabled.
hintsFound	*integer*	number of hints found.
intfsDown	*list of Interface name*	Interfaces which are down.
intfsNoIp	*list of Interface name*	Interfaces without IP address.
intfsNotRouted	*list of Interface name*	Interfaces where routing not enabled.
intfsSparseModeBorderRouter **	*list of Interface name*	Interfaces with PIM sparse-mode and multicast border router enabled. **Since: 4.17.0

2.676 show ip pim config-sanity neighbor [...]

delta	*integer*	time difference.
hintsFound	*integer*	number of hints found.
invalidHello	*list of InvalidHello* see 2.676.1	List of invalid HELLOs.
nfIntf	*list of Interface name*	Neighbor filter enabled interfaces.

2.676.1 InvalidHello

delta	*integer*	Time difference in second between invalid HELLOS.
intfs	*list of Interface name*	interfaces with secondary IP address.

2.677 show ip pim config-sanity sparse-mode [...]

[mroute]	*MrtConfigCheck* see 2.673.1	mroute config check for PIM sparse-mode.
[staticRpSet]	*StaticRpSet* see 2.673.2	staticRp config check for PIM sparse-mode.

2.678 show ip pim interface [...]

interfaces	*table(Interface name) = PimInterface* see 2.678.1	Pim interfaces.

2.678.1 PimInterface

address	*IP address*	Interface address.
[details]	*PimInterfaceDetails* see 2.678.1.1	Pim message detailed counters.
helloPeriod	*integer*	My time interval between sending Hellos in seconds.
intf	*interface name*	Interface name.
[kernelIntf]	*PimKernelInterface* see 2.678.1.2	Interface information from the kernel.
mode	*PimMode* see 2.678.1.3	Pim mode.
neighborCount	*integer*	Number of neighbors on this interface.
[usedDrAddress]	*IP address*	Current DR.
[usedDrPriority]	*integer*	Effective DR priority.

2.678.1.1 PimInterfaceDetails

assertOverrideInterval	*integer*	Assert override interval in seconds.
assertTimeout	*integer*	Assert timeout in seconds.
drPriority	*integer*	My DR priority.
generationId	*integer*	My generation id.
helloDelay	*integer*	My triggered hello delay in seconds.
helloHoldTime	*integer*	My hello hold time in seconds.
joinPruneHoldTime	*integer*	Join/prune hold time in seconds.
joinPrunePeriod	*integer*	Join/prune interval in seconds.
overrideInterval	*integer*	My override interval in milliseconds.
propagationDelay	*integer*	My propagation delay in milliseconds.
[secondaryAddress] **	*list of IP address*	Secondary addresses. **Since: 4.28.1
trackingSupport	*boolean*	Tracking support and disabling join suppression.
usedOverrideInterval	*integer*	Effective propagation delay in milliseconds.
usedPropagationDelay	*integer*	Effective propagation delay in milliseconds.
usedTrackingSupport	*boolean*	Effective tracking support.

2.678.1.2 PimKernelInterface

bytesIn	*integer*	Bytes received.
bytesOut	*integer*	Bytes sent.
name	*string*	Kernel interface name.
pktsDropped	*integer*	Number of packets dropped by kernel.

pktsIn	*integer*	Packets received.
pktsOut	*integer*	Packets received.
pktsQueued	*integer*	Number of packets queued by kernel.
[virtualInterface]	*integer*	Virtual interface number.

2.678.1.3 PimMode

borderRouter	*boolean*	Multicast border router is enabled.
mode	*enum: Bidir* | *Sparse/Bidir* | *None* | *Sparse*	Pim operational mode.

2.679 show ip pim neighbor [...]

revision 2

vrfs	*table(string) = VrfPimNeighbors* see 2.679.1	A mapping between VRF name and PimNeighbors.

2.679.1 VrfPimNeighbors

interfaces	*table(string) = IntfPimNeighbors* see 2.679.1.1	Pim neighbors by interfaces.

2.679.1.1 IntfPimNeighbors

neighbors	*table(IPv4 or IPv6 address) = PimNeighbor* see 2.679.1.1.1	Pim neighbors.

2.679.1.1.1 PimNeighbor

address	*IP address*	Neighbor address.
bfdState	*enum: disabled | down | up | init*	BFD session state with neighbor.
creationTime	*floating point number*	UTC time when neighbor was created.
[detail] **	*boolean*	Display detailed neighbor information. **Since: 4.28.1
holdTime	*integer*	Holdtime for neighbor in seconds.
intf	*interface name*	Interface through which neighbor was discovered.
lastRefreshTime	*floating point number*	UTC time when neighbor last sent Hello message.
mode	*PimMode* see 2.678.1.3	Interface mode.
[multipathColor] **	*integer*	Multipath Color. **Since: 4.28.1
[routerId] **	*IPv4 address*	Router ID. **Since: 4.28.1
[secondaryAddress] **	*list of IP address*	Pim neighbor secondary addresses. **Since: 4.28.1
transport	*enum: tcp | datagram | sctp*	The transport mechanism used for sending/receiving or join/prune messages.

2.680 show ip pim neighbor bfd [...]

revision 2

vrfs	*table(string) = VrfPimBfdNeighbors* see 2.680.1	A mapping between VRF name and PimBfdNeighbors.

2.680.1 VrfPimBfdNeighbors

interfaces	*table(string) = IntfPimNeighbors* see 2.679.1.1	Pim neighbors by interfaces.

2.681 show ip pim protocol counters [...]

counters	*table(string) = PimMessageCounter* see 2.681.1	Map from message type to its counters.
[details]	*PimMessageCounters-Details* see 2.681.2	Detailed interface information.

2.681.1 PimMessageCounter

rxFiltered	*integer*	Filtered received packets.
rxInvalid	*integer*	Invalid/errored received packets.
rxPackets	*integer*	Received packets.
txFiltered	*integer*	Filtered sent packets.
txInvalid	*integer*	Invalid/errored sent packets.
txPackets	*integer*	Sent packets.

2.681.2 PimMessageCountersDetails

subtypeCounters	*table(string) = PimMessageSubtypeCounters* see 2.681.2.1	Map from message type to its subtype counters.

2.681.2.1 PimMessageSubtypeCounters

subtypeCounter	*table(string) = PimMessageCounter* see 2.681.1	Subtype message counters.

2.682 show ip pim register-source [...]

intf	*interface name*	Interface name.
setting	*enum: no config | default | iif | interface*	Register source setting configured.

2.683 show ip pim rp [...]

revision 2

[bidirectional]	*PimRpCandidates* see 2.23	RP Candidate Set for Bidirectional PIM.
[sparseMode]	*PimRpCandidates* see 2.23	RP Candidate Set for Sparse-mode PIM.

2.684 show ip pim rp-candidate [...]

crps	*table(IPv4 or IPv6 address) = PimRpCandidate* see 2.684.1	Map of Candidate RP local to this router.

2.684.1 PimRpCandidate

holdTime	*integer*	C-RP Holdtime.
prefixOptions	*table(IPv4 or IPv6 address prefix) = PimCrpPrefixOptions* see 2.684.1.2	List of priority associated with prefixes.
prefixSet	*table(IPv4 or IPv6 address prefix) = PimCrpPrefix* see 2.684.1.1	List of group prefixes supported by C-RP.
rpAddress	*IP address*	IP address of RP.

2.684.1.1 PimCrpPrefix

prefix	*IP address prefix*	Group Prefix.

2.684.1.2 PimCrpPrefixOptions

prefix	*IP address prefix*	Group Prefix.
priority	*integer*	Group RP Priority.

2.685 show ip pim rp-hash GROUP_ADDR [...]

[bidirectional]	*PimRpHash* see 2.24	RP Hash for Bidirectional PIM.
[sparseMode]	*PimRpHash* see 2.24	RP Hash for Sparse-mode PIM.

2.686 show ip pim upstream joins [...]

see 2.25

2.687 show ip prefix-list NAME detail

ipPrefixLists	*table(string) = IpPrefixList* see 2.687.1	Map prefix list name to prefix list.

2.687.1 IpPrefixList

[duplicateHandling] **	*string*	Enable handling of duplicate entries in URL prefix list. **Since: 4.18.0
ipPrefixEntries	*list of IpPrefixEntry* see 2.687.1.1	Prefix list entries sorted by increasing sequence number.
[ipPrefixListSource] **	*string*	Source URL of the prefix list. **Since: 4.18.0
[vrfName] **	*string*	Enable handling of custom VRF in URL prefix list. **Since: 4.30.1

2.687.1.1 IpPrefixEntry

[eq]	*integer*	Exact prefix length.
filterType	*enum: permit | deny*	Filter type.
[ge]	*integer*	Minimum prefix length.
[le]	*integer*	Maximum prefix length.
prefix	*IP address prefix*	Subnet upon which command filters routes.
seqno	*integer*	Sequence number of prefix list entry.

2.688 show ip prefix-list NAME summary

[entryCount]	*integer*	Prefix list entry count.
[pfxListSource]	*string*	Prefix list source.
[sourceTimeStamp]	*floating point number*	Prefix list source timestamp.

2.689 show ip prefix-list [...]

see 2.687

2.690 show ip rip access-list [...]

see 2.53

2.691 show ip route (PREFIX | ADDR) longer-prefixes [...]

revision 4

Could also be an instance of:
VrfLdpIpRoutes 2.691.1.1.3

vrfs	*table(string) = IpRoutes* see 2.691.1	Route info of each Vrf.

2.691.1 IpRoutes

allRoutesProgrammedHardware	*boolean*	True if every route is programmed in hardware.
allRoutesProgrammedKernel	*boolean*	True if every route is programmed in kernel.
defaultRouteState	*enum: notReachable | reachable | notSet*	The state of the default route.
routes	*table(string) = IpRoute* see 2.691.1.1	List of routes keyed by prefix.
routingDisabled	*boolean*	True if routing is disabled.

2.691.1.1 IpRoute

[cbfVias]	*list of CbfVia* see 2.691.1.1.1	List of Class based forwarding vias.
directlyConnected	*boolean*	True if this route is directly connected though an interface, internal, or is a drop route.
hardwareProgrammed	*boolean*	True if this route is programmed in hardware.
kernelProgrammed	*boolean*	True if this route is programmed in the kernel.
[metric]	*integer*	Used by routing protocol to determine whether one route should be chosen over another. Lower metrics have precedence.
[preference]	*integer*	Administrative distance; lower values have precedence.
routeAction	*enum: drop | forward*	What kind of route it is.
routeLeaked	*boolean*	True if this route is leaked from different VRF.

routeType	*enum: OSPF | connected | dropRoute | iBGP | ospfNssaExternalType1 | vcs | static | CBFLeaked | ospfExternalType2 | routeCacheConnected | OSPFv3 | kernel | martian | ospfInterArea | RIP | dynamicPolicy | bgpAggregate | gribi | BGP | ospfExternalType1 | ISISLevel2 | ospfNssaExternalType2 | ospfSummary | eBGP | nexthopGroupStaticRoute | ISISLevel1*	The IPv4 route type.
[srTePolicy]	*list of RouteSrTePolicy* see 2.691.1.1.1.2	SR-TE policy.
[srcVrf]	*string*	Name of the VRF route is leaked from.
vias	*list of IpRouteVia* see 2.691.1.1.2	List of vias.

2.691.1.1.1 CbfVia

[coloredEndpoints]	*list of CbfColoredTep* see 2.691.1.1.1.1	CBF Endpoint and Color Info.
[dscpValues]	*list of integer*	List of DSCP values.
[labelStack]	*list of integer*	MPLS label stack (top-of-stack label first).
[mplsLabel]	*integer*	MPLS label (service label); may also be included in the labelStack.
[srTePolicy]	*RouteSrTePolicy* see 2.691.1.1.1.2	SR-TE policy.
[tcValues] **	*list of integer*	List of traffic class values. **Since: 4.29.0
[tunnelSource]	*enum: RSVP | VXLAN VTEP Encap | Nexthop | IS-IS SR | EosSdk sBFD | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | RSVP FRR | RSVP LER | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel Type.
[vrfSelectionInfo] **	*VrfSelectionInfo* see 2.691.1.1.1.3	VRF Selection Info. **Since: **currentRelease**

2.691.1.1.1.1 CbfColoredTep

color	*integer*	Color assigned to Tunnel.
endpoint	*IP address prefix*	Tunnel endpoint.

2.691.1.1.1.2 RouteSrTePolicy

color	*integer*	SR-TE policy color.
endpoint	*IP address*	SR-TE policy destination.

2.691.1.1.1.3 VrfSelectionInfo

policyName	*string*	VRF Selection Policy Name.
vrfName	*string*	VRF name to which the route is leaked.

2.691.1.1.2 IpRouteVia

Could also be an instance of:
IpRibRouteVia 2.691.1.1.3

[decapActionType] **	*enum: none | ipip*	Decapsulation type. **Since: 4.27.0
[egressVrf] **	*string*	Name of the nexthop's VRF if the route is an inter-vrf route. **Since: 4.23.0
[fallbackVrf] **	*string*	Fallback VRF. **Since: 4.27.2
[interface]	*interface name*	Name of the interface.
[interfaceDescription]	*string*	Interface description.
[labelStack] **	*list of integer*	MPLS label stack (top-of-stack label first). **Since: 4.20.0
[localAddress] **	*IP address*	Local address of the BFD session when this route via is BFD tracked. **Since: 4.30.1
[localInterface] **	*interface name*	Name of the source interface. **Since: 4.26.0
[mplsLabel]	*integer*	MPLS label (service label); may also be included in the labelStack.
[nexthopAddr]	*IP address*	Nexthop address of the via.
[nexthopGroup] **	*IpRouteNexthopGroup* see 2.691.1.1.2.1	Nexthop group. **Since: 4.21.3
[resolvedVias] **	*list of IpRouteVia* see 2.691.1.1.2	List of resolution vias **Since: 4.26.2
[routerMac]	*MAC address*	VXLAN remote VTEP mac address.
[srv6Sid] **	*IPv6 address*	SRv6 SID (service SID). **Since: 4.29.0
[tunViaBackupVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel backup vias. **Since: 4.25.2
[tunnelDescriptor] **	*TunnelInfo* see 2.151.1.3.1	Tunnel information. **Since: 4.20.5

[viaType]	*enum: backup | active*	Type of Via. PIC has backup Vias.
[viaWeight] **	*string*	UCMP load-balance weight. **Since: 4.20.5
[vias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel vias. **Since: 4.20.5
[vni]	*integer*	VXLAN Virtual Network Identifier.
[vtepAddr]	*IP address*	VXLAN remote VTEP IP address.

2.691.1.1.2.1 IpRouteNexthopGroup

groupId	*integer*	Nexthop group Id.
groupName	*string*	Nexthop group name.

2.691.1.1.3 IpRibRouteVia

[decapActionType] **	*enum: none | ipip*	Decapsulation type. **Since: 4.27.0
[egressVrf] **	*string*	Name of the nexthop's VRF if the route is an inter-vrf route. **Since: 4.23.0
[fallbackVrf] **	*string*	Fallback VRF. **Since: 4.27.2
[interface]	*interface name*	Name of the interface.
[interfaceDescription]	*string*	Interface description.
[labelStack] **	*list of integer*	MPLS label stack (top-of-stack label first). **Since: 4.20.0
[localAddress] **	*IP address*	Local address of the BFD session when this route via is BFD tracked. **Since: 4.30.1
[localInterface] **	*interface name*	Name of the source interface. **Since: 4.26.0
[mplsLabel]	*integer*	MPLS label (service label); may also be included in the labelStack.
[nexthopAddr]	*IP address*	Nexthop address of the via.
[nexthopGroup] **	*IpRouteNexthopGroup* see 2.691.1.1.2.1	Nexthop group. **Since: 4.21.3
[nexthopName]	*string*	Nexthop name.
[resolvedVias] **	*list of IpRouteVia* see 2.691.1.1.2	List of resolution vias **Since: 4.25.2
[routerMac] **	*MAC address*	VXLAN remote VTEP mac address. **Since: 4.18.0
[srv6Sid] **	*IPv6 address*	SRv6 SID (service SID). **Since: 4.29.0
[tunViaBackupVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel backup vias. **Since: 4.25.2
[tunnelDescriptor] **	*TunnelInfo* see 2.151.1.3.1	Tunnel information. **Since: 4.20.5
[viaType]	*enum: backup | active*	Type of Via. PIC has backup Vias.
[viaWeight] **	*string*	UCMP load-balance weight. **Since: 4.20.5

`[vias]` **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel vias. **Since: 4.20.5
`[vni]` **	*integer*	VXLAN Virtual Network Identifier. **Since: 4.18.0
`[vtepAddr]` **	*IP address*	VXLAN remote VTEP IP address. **Since: 4.18.0
`vrfs`	*table(string) = IpRoutes* see 2.691.1	Route info of each Vrf.

2.692 show ip route [...]

revision 4

see 2.691

2.693 show ip route fec [...]

`vrfs`	*table(string) = FecRoutesModel* see 2.693.1	Mapping of VRF name to its FECs.

2.693.1 FecRoutesModel

`fecRoutes`	*table(string) = FecRouteModel* see 2.693.1.1	A mapping of FEC ID to its routes.

2.693.1.1 FecRouteModel

`ipv4PrefixCount`	*integer*	Number of IPv4 prefixes using this FEC.
`ipv6PrefixCount`	*integer*	Number of IPv6 prefixes using this FEC.
`nextHops`	*list of FecNextHopModel* see 2.693.1.1.1	Next hops for this FEC.
`[orderedFec]`	*boolean*	This is an ordered FEC.
`[ucmpEligibleFec]` **	*boolean*	This is an UCMP eligible FEC. **Since: **currentRelease**

2.693.1.1.1 FecNextHopModel

`[decapActionType]` **	*enum: none | ipip*	Decapsulation type. **Since: 4.27.0
`[directlyConnected]`	*boolean*	This is a direct connection.
`[fallbackVrf]` **	*string*	Fallback VRF name. **Since: 4.27.2
`[groupID]`	*integer*	The ID of the next hop group.
`[groupName]`	*string*	The name of the next hop group.
`[hop]`	*IP address*	The hop prefix.
`[internalConnection]`	*boolean*	The connection is internal.
`[intfID]`	*interface name*	The Interface ID.
`[isDynamicTunnel]`	*boolean*	This route is a dynamic tunnel connection.
`[tunnelID]`	*integer*	The ID of the tunnel connection.

[tunnelTypeStr]	*enum: staticTunnel* | *rsvpLerSubTunnel* | *vxlanVtepEncapTunnel* | *rsvpFrrTunnel* | *srTePolicyTunnel* | *bgpLuTunnel* | *nexthopGroupTunnel* | *eosSdkSbfdSessionTunnel* | *bgpLuForwardingTunnel* | *nexthopTunnel* | *maxTunnelType* | *staticV6Tunnel* | *rsvpTunnel* | *vxlanTunnel* | *evpnVxlanTunnel* | *isisFlexAlgoTunnel* | *staticInterfaceTunnel* | *staticMcastTunnel* | *srTunnel* | *invalidTunnel* | *tiLfaTunnel* | *staticV4Tunnel* | *ldpTunnel* | *rsvpLerTunnel* | *ipInIpTunnel* | *srv6TransportTunnel* | *srV4Tunnel* | *srTeSegmentListTunnel* | *srV6Tunnel* | *mldpP2mpTunnel*	The type of tunnel connection.
[viaNull]	*boolean*	There are no next hops.

2.694 show ip route fec protocol summary

protocols	*table(string) = FecTypeBreakdown* see 2.694.1	Breakdown of FECs per FEC source.
receiveFecCount	*integer*	Count of receive FECs.

2.694.1 FecTypeBreakdown

drop	*integer*	Number of FECs of type drop.
ecmp	*integer*	Number of FECs with multiple next hops.
nonEcmp	*integer*	Number of FECs with single next hop.
stale	*integer*	Number of stale FECs.

2.695 show ip route fec summary [...]

fecBreakdowns	*table(numeric string) = FecBreakdown* see 2.695.1	Breakdown of FECs, Prefixes and Hierarchical References per FEC Size.

2.695.1 FecBreakdown

fecSize	*integer*	FEC Size.
numFecs	*integer*	Total number of FECs.
numHierarchicalRefs **	*integer*	Total number of hierarchical references. **Since: 4.20.5
numPrefixes	*integer*	Total number of prefixes.

2.696 show ip route host [...]

revision 2

vrfs	*table(string) = IpRoutesHost* see 2.696.1	A mapping of VRF to its host routes.

2.696.1 IpRoutesHost

allRoutesProgrammedHardware	*boolean*	Every route is programmed in hardware.
routes	*table(string) = IpRouteHost* see 2.696.1.1	IP host routes keyed by prefix.
[routingDisabled]	*boolean*	Routing is disabled.

2.696.1.1 IpRouteHost

hardwareProgrammed	*boolean*	True if this route is programmed in hardware.
routeType	*enum: directedBcast | FIB | attached | broadcast | receive*	The ip route host type.
vias	*list of IpRouteHostVia* see 2.696.1.1.1	List of host vias.

2.696.1.1.1 IpRouteHostVia

[interface]	*interface name*	Name of the interface.
[nexthopAddr]	*IP address*	Nexthop address of the via.
[routerMac]	*MAC address*	VXLAN remote VTEP mac address.
toCpu	*boolean*	True if route goes to cpu.
[vni]	*integer*	VXLAN Virtual Network Identifier.
[vtepAddr]	*IP address*	VXLAN remote VTEP IP address.

2.697 show ip route kernel unprogrammed [...]

vrfs	*table(string) = UnprogrammedRoutes* see 2.697.1	A dictionary of unprogrammed routes keyed by VRF.

2.697.1 UnprogrammedRoutes

prefixes	*table(IPv4 or IPv6 address prefix) = UnprogrammedRoute* see 2.697.1.1	A dictionary of unprogrammed route details keyed by prefix.

2.697.1.1 UnprogrammedRoute

errorCode	*integer*	Standard error code.
[interface]	*interface name*	Unprogrammed route interface.

2.698 show ip route multicast [...]

revision 3

routes	*table(IPv4 or IPv6 address prefix) = MulticastRibRoute* see 2.698.1	Multicast static routes.

2.698.1 MulticastRibRoute

[routeCreationTime]	*string*	Route creation time.
routeMetric	*integer*	Route metric.
routePreference	*integer*	Route preference.
routeProtocol	*string*	Routing protocol through which this route was learnt.
vias	*table(IPv4 or IPv6 address) = string*	set of next-hop addresses that this route is reachable via.

2.699 show ip route next-hop nexthop-group [...]

revision 4

see 2.691

2.700 show ip route next-hop vxlan [...]

revision 4

see 2.691

2.701 show ip route rpf unicast [...]

[interfaces]	*table(Interface name) = UrpfInterface* see 2.701.1	Mapping of interface to the additional interface state details.
[prefix]	*IP address prefix*	Destination Prefix.

[viaDefaultRoute]	*boolean*	uRPF is carried out via the default route.

2.701.1 UrpfInterface

allowDefault	*boolean*	Packets are permitted to come in on the default interface.
[lookupVrf] **	*string*	Name of uRPF lookup VRF. **Since: 4.28.2
uRpfMode	*enum: loose | strictDefault | disable | strict*	URPF mode state.

2.702 show ip route static not-installed bfd [...]

vrfs	*table(string) = NotInstalledBfd* see 2.702.1	Route info of each Vrf.

2.702.1 NotInstalledBfd

routes	*table(string) = AllRoutes* see 2.702.1.1	Bfd disabled static routes.

2.702.1.1 AllRoutes

viaGroups	*list of ViaList* see 2.702.1.1.1	List of all paths.

2.702.1.1.1 ViaList

[metric]	*integer*	Used by routing protocol to determine whether one route should be chosen over another. Lower metrics have precedence.
[preference]	*integer*	Administrative distance; lower values have precedence.
vias	*list of IpRouteSimpleVia* see 2.702.1.1.1.1	List of vias.

2.702.1.1.1.1 IpRouteSimpleVia

Could also be an instance of:
IpRibRouteVia 2.691.1.1.3
IpRouteVia 2.691.1.1.2

[egressVrf] **	*string*	Name of the nexthop's VRF if the route is an inter-vrf route. **Since: 4.23.0
[interface]	*interface name*	Name of the interface.
[localAddress] **	*IP address*	Local address of the BFD session when this route via is BFD tracked. **Since: 4.30.1
[nexthopAddr]	*IP address*	Nexthop address of the via.

2.703 show ip route summary [...]

revision 3

protoModelStatus **	*ServiceRoutingProtoModelStatus* see 2.703.2	Active and configured protocol models. **Since: 4.23.0
vrfs	*table(string) = IpRouteSummaryForVrf* see 2.703.1	Route summary of each VRF.

2.703.1 IpRouteSummaryForVrf

aggregate	*integer*	Number of aggregate routes.
attached	*integer*	Number of attached routes.
bgpCounts	*bgpTotals* see 2.703.1.1	BGP route count breakdown.
connected	*integer*	Number of connected routes.
dynamicPolicy **	*integer*	Number of dynamic policy routes. **Since: 4.21.0
gribi **	*integer*	Number of gRIBI routes. **Since: 4.26.2
internal	*integer*	Number of internal routes.
isisCounts	*isisTotals* see 2.703.1.2	IS-IS route count breakdown.
maskLen	*table(numeric string) = integer*	Counts for each mask length.
ospfCounts	*ospfTotals* see 2.703.1.3	OSPF route count breakdown.
ospfv3Counts	*ospfv3Totals* see 2.703.1.4	OSPFv3 route count breakdown.
rip	*integer*	Number of rip routes.
static	*integer*	Number of static routes.
staticNexthopGroup	*integer*	Number of static nexthop-group routes.
staticNonPersistent	*integer*	Number of static routes configured using the SDK with persistent=False.
staticPersistent	*integer*	Number of static routes configured to persist.
totalRoutes	*integer*	Total number of routes.
[vcs]	*integer*	Number of VXLAN Control Service routes.
[vrfCount]	*integer*	Total number of vrf instances.

2.703.1.1 bgpTotals

bgpExternal	*integer*	Number of bgp External routes.
bgpInternal	*integer*	Number of bgp Internal routes.
bgpLocal **	*integer*	Number of bgp Other routes. **Since: 4.22.1
bgpTotal	*integer*	Total number of bgp routes.

2.703.1.2 isisTotals

isisLevel1	*integer*	Number of isis Level-1 routes.
isisLevel2	*integer*	Number of isis Level-2 routes.
isisTotal	*integer*	Total number of isis routes.

2.703.1.3 ospfTotals

nssaExternal1	*integer*	Number of Ospf NSSA External-1 routes.
nssaExternal2	*integer*	Number of Ospf NSSA External-2 routes.
ospfExternal1	*integer*	Number of Ospf External-1 routes.
ospfExternal2	*integer*	Number of Ospf External-2 routes.
ospfInterArea	*integer*	Number of Ospf Inter-area routes.
ospfIntraArea	*integer*	Number of Ospf Intra-area routes.
ospfTotal	*integer*	Total number of Ospf routes.

2.703.1.4 ospfv3Totals

ospfv3External1	*integer*	Number of Ospfv3 External-1 routes.
ospfv3External2	*integer*	Number of Ospfv3 External-2 routes.
ospfv3InterArea	*integer*	Number of Ospfv3 Inter-area routes.
ospfv3IntraArea	*integer*	Number of Ospfv3 Intra-area routes.
ospfv3NssaExternal1	*integer*	Number of Ospfv3 NSSA External-1 routes.
ospfv3NssaExternal2	*integer*	Number of Ospfv3 NSSA External-2 routes.
ospfv3Total	*integer*	Total number of Ospfv3 routes.

2.703.2 ServiceRoutingProtoModelStatus

configuredProtoModel	*enum: multi-agent | ribd*	Configured service routing protocols model.
operatingProtoModel	*enum: multi-agent | ribd*	Active service routing protocols model.

2.704 show ip route vrf all summary brief [...]

see 2.703.1

2.705 show ip security applied-profile [...]

ipsecAppliedProfiles	*table(string) = IpsecAppliedProfile* see 2.705.1	A mapping between Ipsec profile name and profile applied to aninterface.

2.705.1 IpsecAppliedProfile

appliedIntf	*list of Interface name*	Tunnel interface List for this profile.
appliedPath **	*list of string*	Path list for this profile. **Since: 4.22.1
ipsecConnectionDetails	*table(string) = IpsecConnectionDetail* see 2.705.1.1	A mapping between Ipsec profile info and corresponding connection.

2.705.1.1 IpsecConnectionDetail

connectionName	*string*	Name of the IPsec connection.
dstAddr	*IP address*	Destination IP Address.
pathDict **	*table(string) = string*	Ipsec path dict. **Since: 4.22.1
srcAddr	*IP address*	Source IP Address.
tunnelDict **	*table(Interface name) = string*	Tunnel interface dict, with SA info. **Since: 4.20.5
tunnelNs	*string*	Tunnel namespace for this profile.

2.706 show ip security connection [...]

revision 2

connections	*table(string) = IpsecConnection* see 2.706.1	A mapping between tunnel name andIPsec tunnel.

2.706.1 IpsecConnection

connName	*string*	Name of the IPsec connection.
daddr	*IP address*	Destination IP Address.
dynPathDict **	*table(string) = IpsecDynamicPathInfoM* see 2.706.1.1	A mapping of paths to dynamic IPsec info. **Since: 4.29.2
dynamicPath **	*boolean*	Identifies if peer is a dynamic path. **Since: 4.29.2
[inboundSA]	*SecurityAssociation* see 2.706.1.2	SA for an inbound Flow.
initiator **	*boolean*	This peer is an initiator. **Since: 4.29.2
ipsecInputDataBytes	*integer*	Ipsec Input Data in bytes.
ipsecInputDataPkts	*integer*	Ipsec Input Data in packets.
ipsecOutputDataBytes	*integer*	Ipsec Output Data in bytes.
ipsecOutputDataPkts	*integer*	Ipsec Output Data in packets.
[outboundSA]	*SecurityAssociation* see 2.706.1.2	SA for an outbound Flow.
[pathDict] **	*table(string) = string*	Path name for this connection. **Since: 4.22.1
rekeyTime	*integer*	Seconds remaining until next rekey.
remoteIp **	*IP address*	Remote IP. **Since: 4.29.2
saddr	*IP address*	Source IP Address.
[tunnelDict]	*table(Interface name) = string*	Tunnel interface List for this connection.
tunnelNs	*string*	Tunnel namespace for this profile.
upTimeStr	*string*	Ipsec connection uptime.
vrfName **	*string*	VRF name. **Since: 4.29.2

2.706.1.1 IpsecDynamicPathInfoM

dhCryptoInfo	*list of IpsecDynamicSpiInfoM* see 2.706.1.1.3	DH crypto entries.
dhStateInfo	*list of IpsecDynamicDhStateM* see 2.706.1.1.1	DH state entries.
lastEstablishedTime	*floating point number*	Time when session was last established.
lastKnownLocalRekeyCount	*integer*	Last rekey count.

pathState	*enum: connected* | *established* | *dhInitiatorInSaInstalled* | *dhResponderOldSaDeleted* | *saResponderInSaPkts-Received* | *dhInitiatorRekeyComplete* | *saResponderInSaInstalled* | *saInitiatorInSaInstalled* | *saInitiatorNonceRequest-Received* | *dhInitiatorInSaPkts-Received* | *saInitiatorSimultaneous* | *dhResponderRekey-Complete* | *saInitiatorRekeyComplete* | *saResponderNonce-ResponseSent* | *initial* | *dhInitiatorOldSaDeleted* | *saResponderOutSaInstalled* | *dhInitiatorSimultaneous* | *dhInitiatorRekeyInitiated* | *dhInitiatorOutSaInstalled* | *saResponderSimultaneous* | *dhResponderSimultaneous* | *saResponderRekeyComplete* | *invalid* | *dhResponderRekeyInitiated* | *saInitiatorNonceRequest-Sent* | *saResponderNonceRequest-Received* | *dhResponderOutSaInstalled* | *dhResponderInSaInstalled* | *saInitiatorOutSaInstalled* | *saInitiatorNonceResponse-Received*	Dynamic path state.
saCryptoInfo	*list of IpsecDynamicSpiInfoM* see 2.706.1.1.3	SA crypto entries.
saInfo	*list of IpsecDynamicSaInfoM* see 2.706.1.1.2	SA info entries.
saSmState	*enum: invalid* | *simultaneous* | *dhRekeyInProgress* | *responder* | *notRekeying* | *initiator*	Current SA state.
sessionPktsIn	*integer*	Session input packets.
sessionPktsOut	*integer*	Session output packets.

2.706.1.1.1 IpsecDynamicDhStateM

dhRekeyState	*enum: inSaInstalled | invalid | inProgress | notRekeying | peerOutSaInstalled | initiated*	Current DH rekey state.
dhSmState	*enum: invalid | simultaneous | dhRekeyInProgress | responder | notRekeying | initiator*	Current DH state machine state.
localRekey	*integer*	Local rekey count.
peerRekey	*integer*	Peer rekey count.

2.706.1.1.2 IpsecDynamicSaInfoM

addTime	*floating point number*	Time when the SA was programmed.
initiator	*boolean*	This peer is an initiator.
pairSpi	*integer*	Pair SPI.
peerNonceType	*enum: invalid | response | request | init*	Peer nonce type.
pkts	*integer*	Packets seen on this path.
saRekeyState	*enum: initial | installed | peerOutSaInstalled | invalid | created*	SA rekey state.
saType	*enum: newDhOutSa | inSa | outSa | oldInSa | oldOutSa | inOutSa | newDhInSa | newSaOutSa | newSaInSa | unknownSa*	SA type.
spi	*integer*	SPI value.

2.706.1.1.3 IpsecDynamicSpiInfoM

inSpi	*integer*	Inbound SPI value.
localRekey	*integer*	Local rekey count.
outSpi	*integer*	Outbound SPI value.
peerRekey	*integer*	Peer rekey count.

2.706.1.2 SecurityAssociation

daddr	*IP address*	Dest IP Address.
family	*enum: unknownFamily | ipv4 | ipv6*	v4 or v6 family.
lifetimeCfg	*LifetimeCfg* see 2.706.1.2.1	Lifetime interval config in bytes/time:.
lifetimeCurrent	*LifetimeCur* see 2.706.1.2.2	Lifetime interval current status in bytes/time:.

mode	*enum: inboundTrigger | routeOptimization | unknownMode | beet | transport | tunnel*	mode of SA (tunnel, transport, beet, pass etc).
packetsOutOfReplayWindow	*integer*	Number of packets outside Replay Window.
proto	*enum: unknownProto | ah | esp*	IP Transform Protocol (AH/ESP).
replayErrors	*integer*	Number of Replay Errors.
replayWindow	*integer*	Sequence number window for replay protection.
reqid	*integer*	Request Id for a given SA pair.
saddr	*IP address*	Source IP Address.
seq	*integer*	Sequence Number to help with replay protection.
spi	*integer*	Security Policy Index for ESP/AH.
statsIntegrityFailed	*integer*	Number of Integrity check failures for incoming packets.

2.706.1.2.1 LifetimeCfg

xfrmPolicyByteLimitExpiry	*integer*	Expiry of byte transfer limit.
xfrmPolicyByteLimitWarning	*integer*	Warning about impending expiry of byte transfer limit.
xfrmPolicyPacketLimitExpiry	*integer*	Expiry of packet transfer limit.
xfrmPolicyPacketLimit-Warning	*integer*	Warning about impending expiry of packet transfer limit.
xfrmPolicyTimeLimitExpiry	*integer*	Expiry of time limit (in secs).
xfrmPolicyTimeLimitWarning	*integer*	Warning about impending expiry of time limit (in secs).
xfrmPolicyUseTimeExpiry	*integer*	Expiry of use time.
xfrmPolicyUseTimeWarning	*integer*	Warning about impending expiry of use time.

2.706.1.2.2 LifetimeCur

bytesProcessed	*integer*	Number of bytes processed in tx & rx.
packetsProcessed	*integer*	Number of pkts processed in tx & rx.
policyAddTime	*integer*	Timestamp when the policy was added.
policyLastAccessTime	*integer*	Timestamp when policy was last accessed.

2.707 show ip security daemon [...]

strongSwanEnabled	*boolean*	strongSwan enabled.
strongSwanRunning	*boolean*	strongSwan running.

2.708 show ip security daemon log [...]

`[daemonLog]`	*list of string*	Log in charon.log.

2.709 show ip security fips status

`fipsSelfTestStatus`	*enum: passed* | *failed* | *unknown*	FIPS self-test status.
`ikeFipsEnabled`	*boolean*	Control plane FIPS enabled.
`platformFipsEnabled`	*boolean*	Data path FIPS enabled.

2.710 show ip security key controller local

`[keyController]`	*IpsecLocalKey* see 2.710.1	Configured IPsec Key Controller.

2.710.1 IpsecLocalKey

`dhGroup`	*enum: modp3072* | *modp2048* | *modp4096* | *modp2048s256* | *ecp384* | *modp1536* | *modp1024* | *ecp521* | *unknownDhGroup* | *modp6144* | *modp768*	Diffie-Hellman (DH) group number.
`encryptAlgo`	*enum: nullesp* | *des* | *aes256* | *aes256gcm128* | *unknownEncr* | *aes128gcm64* | *aes128gcm128* | *aes128*	Encryption Algorithm.
`firstDhKeyAddTime` **	*integer*	Seconds the initial DH key was created. **Since: 4.25.2
`firstKeyAddTime`	*floating point number*	Seconds the initial DH key was created.
`hashAlgo`	*enum: sha256* | *unknownIntegrity* | *sha512* | *sha384* | *sha1* | *nullhash*	Integrity Algorithm.
`lastKeyUptime`	*integer*	Seconds since the last key was created.
`rekeyTime`	*integer*	Seconds remaining until next rekey.

2.711 show ip security key controller peer [...]

`peers`	*table(IPv4 or IPv6 address) = IpsecPeer* see 2.711.1	A mapping between peer address and IPsec peer info of the key controller.

2.711.1 IpsecPeer

dhGroup	*integer*	Diffie-Hellman (DH) group number.
encryptAlgo	*enum: nullesp* | *des* | *aes256* | *aes256gcm128* | *unknownEncr* | *aes128gcm64* | *aes128gcm128* | *aes128*	Encryption Algorithm.
hashAlgo	*enum: sha256* | *unknownIntegrity* | *sha512* | *sha384* | *sha1* | *nullhash*	INTEGRITY Algorithm.
initiator	*boolean*	The peer is the initiator.
lastKeyUptime	*integer*	Seconds since the last key material received.
peerIp	*IP address*	Peer IP Address.
peerStatus	*enum: peerUp* | *dhGroupMismatch* | *unknownPeerState* | *nonceLenMismatch* | *licenseDisabled* | *encryptAlgoMismatch* | *cryptoKeyGenerationErr* | *dhLenMismatch* | *hashAlgoMismatch*	Status of the IPsec key controller peer.
uptime	*floating point number*	Seconds since the peer came up.

2.712 show ip security policy [...]

ipsecPolicies	*table(string) = IpsecPolicy* see 2.712.1	A mapping between IPSEC policy name and an IPSEC policy.

2.712.1 IpsecPolicy

dhGroup	*enum: modp3072* | *modp2048* | *modp4096* | *modp2048s256* | *ecp384* | *modp1536* | *modp1024* | *ecp521* | *modp6144* | *modp768*	Identifier for Diffie-Hellman (DH) group.
encryption	*enum: nullesp* | *des* | *aes256* | *aes256gcm128* | *unknownEncr* | *aes128gcm64* | *aes128gcm128* | *aes128*	Encryption Algorithm used in IKE.
ikeLifetime	*integer*	Lifetime for the IKE SA in hours.
integrity	*enum: sha256* | *unknownIntegrity* | *sha512* | *sha384* | *sha1* | *nullhash*	Hash algorithm to verify integrity mechanism.
ipsecAuth	*enum: rsa_sig* | *unknownAuth* | *pre_share*	Peer Authentication method.
rekey	*boolean*	Rekey Enabled.

version	*string*	Version of IKE Policy.

2.713 show ip security profile [...]

ipsecProfiles	*table(string) = IpsecProfile* see 2.713.1	A mapping between a IPSEC profile name and an IPSEC profile.

2.713.1 IpsecProfile

policyName	*string*	Unique identifier of IKE Policy.
[securityAssociation]	*string*	Transformset associated with profile.

2.714 show ip security security-association [...]

ipsecSecurityAssociations	*table(string) = IpsecSecurityAssociation* see 2.714.1	A mapping between a security association and it's information.

2.714.1 IpsecSecurityAssociation

byteLimit **	*integer*	Byte count limit for the SA. **Since: 4.22.0
espAes	*enum: nullesp | des | aes256 | aes256gcm128 | unknownEncr | aes128gcm64 | aes128gcm128 | aes128*	ESP Advanced Encryption Standard Algorithm.
espSha	*enum: sha256 | unknownIntegrity | sha512 | sha384 | sha1 | nullhash*	ESP Secure Hash Algorithm.
packetLimit **	*integer*	Packet count limit for the SA. **Since: 4.22.0
pfsGroup	*enum: modp3072 | modp2048 | modp4096 | modp2048s256 | ecp384 | modp1536 | modp1024 | ecp521 | modp6144 | modp768*	PFS Group to force a new DH key exchange.
replayWindowSize **	*integer*	Anti-Replay Window size to detect replay attack. **Since: 4.20.5
saLifetime	*integer*	Lifetime for the SA in hours.

2.715 show ip source binding static [...]

bindingEntries	*list of IpMacBindingEntry* see 2.715.1	List of valid IP-MAC bindings.

2.715.1 IpMacBindingEntry

entryType	*enum: dynamic | static*	Binding type.
interface	*interface name*	Interface name.
ipAddr	*IPv4 address*	IPv4 address.
leaseTime	*integer*	Lease time in seconds.
macAddr	*MAC address*	MAC address.
vlanId	*integer*	Vlan this binding applies to.

2.716 show ip user route [...]

vrfs	*table(string) = IpRibRoutes* see 2.716.1	A mapping from VRF to the RIB routes it contains.

2.716.1 IpRibRoutes

routes	*table(string) = IpRibRoutePreference* see 2.716.1.1	A mapping from prefix to a collection of RIB routes.

2.716.1.1 IpRibRoutePreference

preference	*table(numeric string) = IpRibRoute* see 2.716.1.1.1	A mapping from preference to a RIB route.

2.716.1.1.1 IpRibRoute

[dropNextHopName]	*string*	Nexthop name for drop route.
[nexthopGroupName]	*string*	Nexthop group name.
[preference]	*integer*	Administrative distance; lower values have precedence.
routeType	*enum: drop | nexthopGroup | forward*	RIB route type.
[tag]	*integer*	Route tag number.
vias	*list of IpRibRouteVia* see 2.691.1.1.3	List of vias.

2.717 show ip verify source

interfaceStates	*table(Interface name) = IpsgInterfaceOperational-State* see 2.717.1	A mapping from interfaces to IP source guardoperational states on these interfaces.

2.717.1 IpsgInterfaceOperationalState

operationalState	*enum: IP source guard disabled | Error: hardware resource unavailable | IP source guard enabled*	Operational state of IP source guard on an-interface.

2.718 show ip verify source detail

detailStates	*table(Interface name) = IpsgBindingDetailStateList* see 2.718.1	IP source binding entries and their state.

2.718.1 IpsgBindingDetailStateList

intfDetailList	*list of IpsgBindingDetailState* see 2.718.1.1	IP source binding entries and the states.

2.718.1.1 IpsgBindingDetailState

ipAddr	*IPv4 address*	IPv4 address.
macAddr	*MAC address*	MAC address.
state	*enum: IP source guard disabled | Port-Channel down | Hardware resource unavailable | Active*	Operational state of an IP source binding entry.
vlanId	*integer*	VLAN ID of an IP source binding entry.

2.719 show ip verify source vlan

vlanStates	*table(numeric string) = IpsgVlanOperationalState* see 2.719.1	A mapping from VLANs to IP source guard operationalstates on these VLANs.

2.719.1 IpsgVlanOperationalState

operationalState	*enum: IP source guard disabled | Error: vlan classification failed | Error: hardware resource unavailable*	Operational state of IP source guard on a VLAN.

2.720 show ip virtual-router [...]

revision 2

advertiseInterval	*integer*	Mac address advertisement interval.
virtualMacs	*list of VirtualRouterMacModel* see 2.720.4	Virtual MAC addresses configured.
virtualRouters	*list of VarpRouterBase* see 2.720.1	Virtual Routers configured.

2.720.1 VarpRouterBase

Could also be an instance of:
VarpRouterV4 2.720.2
VarpRouterV6 2.720.3

interface	*interface name*	Name of Interface.
interfaceStatus	*enum: down* | *adminDown* | *notApplicable* | *uninitDown* | *up*	Connection status of the interface.
protocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[state]	*enum: stopped* | *active*	stopped – the virtual router is inactive, active – the virtual router is active.
vrfName	*string*	Name of VRF.

2.720.2 VarpRouterV4

interface	*interface name*	Name of Interface.
interfaceStatus	*enum: down* | *adminDown* | *notApplicable* | *uninitDown* | *up*	Connection status of the interface.
ipAddress	*IpAddrAndMask* see 2.241.1.2.2.9	Interface address.
protocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[state]	*enum: stopped* | *active*	stopped – the virtual router is inactive, active – the virtual router is active.
virtualIps	*list of IpAddrAndMask* see 2.241.1.2.2.9	Virtual IPv4 addresses.
vrfName	*string*	Name of VRF.

2.720.3 VarpRouterV6

interface	*interface name*	Name of Interface.
interfaceStatus	*enum: down* | *adminDown* | *notApplicable* | *uninitDown* | *up*	Connection status of the interface.
ipv6Addresses	*list of Ip6AddrAndMask* see 2.241.1.2.2.8	Ipv6 interface addresses.
protocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[state]	*enum: stopped* | *active*	stopped – the virtual router is inactive, active – the virtual router is active.
virtualIpv6	*list of IPv6 address*	Virtual Ipv6 addresses.
vrfName	*string*	Name of VRF.

2.720.4 VirtualRouterMacModel

Could also be an instance of:
VarpRouterMac 2.720.6
MlagPeerRouterMac 2.720.5

macAddress	*MAC address*	Virtual router MAC address.
macType	*enum: varp* | *mlag*	varp – VARP MAC address, mlag – Mlag peer router MAC address.

2.720.5 MlagPeerRouterMac

configured	*boolean*	True if the routing using Mlag peer MAC address is configured.
macAddress	*MAC address*	Virtual router MAC address.
macType	*enum: varp* | *mlag*	varp – VARP MAC address, mlag – Mlag peer router MAC address.

2.720.6 VarpRouterMac

macAddress	*MAC address*	Virtual router MAC address.
macType	*enum: varp* | *mlag*	varp – VARP MAC address, mlag – Mlag peer router MAC address.
[mask] **	*MAC address*	Virtual router MAC address mask. **Since: 4.21.0
subnetRoutes **	*boolean*	Virtual subnet routes enabled. **Since: 4.24.2

2.721 show ipv6 access-lists [...]

see 2.53.2

2.722 show ipv6 address virtual source-nat sourceNatAddr

`vrfs`	*table(string) = IPv6 address*	A mapping of VRF names to their source NAT IPv6 address.

2.723 show ipv6 bgp (IP | PREFIX) [...]

revision 3

see 2.17

2.724 show ipv6 bgp (installed | not-installed) [...]

revision 3

see 2.17

2.725 show ipv6 bgp [...]

revision 3

see 2.17

2.726 show ipv6 bgp access-list AS_PATH_LIST_NAME [...]

revision 3

see 2.17

2.727 show ipv6 bgp field-set FSNAME [...]

revision 3

see 2.17

2.728 show ipv6 bgp large-community { LARGE_COMM_VALUE } detail exact [...]

revision 3

see 2.17

2.729 show ipv6 bgp peers [...]

revision 2

see 2.532

2.730 show ipv6 bgp peers update errors [...]

revision 2

see 2.533

2.731 show ipv6 bgp regex REGEX [...]

revision 3

see 2.17

2.732 show ipv6 bgp summary [...]

revision 3

see 2.46

2.733 show ipv6 dhcp relay installed routes [...]

`vrfs`	*table(string) = Dhcp6InstalledRoutesPerVrf* see 2.733.1	DHCP Relay prefixes keyed by VRF.

2.733.1 Dhcp6InstalledRoutesPerVrf

`routes`	*table(IPv4 or IPv6 address prefix) = Dhcp6InstalledRoute* see 2.733.1.1	DHCP Relay installed routes keyed by IPv6 prefix.

2.733.1.1 Dhcp6InstalledRoute

`clientIpAddress`	*IPv6 address*	DHCP client's IP address.
`expiryTime`	*floating point number*	Expiry time of prefix in UTC.
`interface`	*interface name*	DHCP client's interface.

2.734 show ipv6 dhcp snooping

enabled	*boolean*	DHCPv6 Snooping is enabled.
enabledVlans	*list of integer*	VLANs on which snooping is configured.
operational	*boolean*	DHCPv6 Snooping is operational.
operationalVlans	*list of integer*	VLANs on which snooping is operational.
remoteIdOption	*boolean*	Insertion of Option-37 is enabled.

2.735 show ipv6 dhcp snooping counters [...]

detail_ **	*boolean*	Counter values in detail. **Since: 4.25.2
enabled	*boolean*	DHCPv6 Snooping is enabled.
vlanCounters	*table(numeric string) = VlanCounters* see 2.542.1	Per-VLAN counter information.

2.736 show ipv6 dhcp snooping counters debug

[dhcpOpErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Dhcp Optype.
[disabledErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Snooping disabled.
enabled	*boolean*	DHCPv6 Snooping is enabled.
[forwardedCounters]	*SnoopingRelayCounters* see 2.543.1	Forwarded.
[infoOptErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Info Option.
[parseErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Parse error.
[receivedCounters]	*SnoopingRelayCounters* see 2.543.1	Recieved.
[remoteIdErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid Remote-ID Option.
[resetTime]	*floating point number*	Last time counters were reset.
[vlanIdErrCounters]	*SnoopingRelayCounters* see 2.543.1	Dropped - Invalid VlanId.

2.737 show ipv6 dhcp snooping hardware

enabled	*boolean*	DHCPv6 Snooping is enabled.
vlans	*list of integer*	VLANs on which snooping is enabled.
vlansPerSlice	*table(string) = VlanList* see 2.544.1	VLANs enabled per slice.

2.738 show ipv6 hardware ale routes [...]

see 2.555

2.739 show ipv6 hardware ale routes vrf all [...]

see 2.556

2.740 show ipv6 hardware fib ecmp resilience [...]

see 2.558

2.741 show ipv6 hardware fib ecmp resilience summary [...]

see 2.559

2.742 show ipv6 hardware fib routes [...]

see 2.562

2.743 show ipv6 interface [...]

interfaces	*table(Interface name) = Ip6Status* see 2.743.1	Maps an interface name to its IPv6 status.

2.743.1 Ip6Status

acceptUnsolicitedNa **	*boolean*	Accept unsolicited neighbor advertisements to update neighbor cache as per RFC9131. **Since: 4.28.2
[addrSource] **	*enum: slaac | manual*	How the address is configured on the interface. **Since: 4.20.0
addresses	*list of IpAddress* see 2.467.1.5.1	Addresses configured on this interface.
autoConfigStatus	*enum: stateful | dhcp | stateless*	IPv6 address assignment option.
dadAttempts	*integer*	Number of DAD attempts.
[dadStatus]	*enum: strict | disabled | enabled | unsupported | unavailable*	Current status of Duplicate Address Detection.
[enhancedDad] **	*boolean*	RFC7527 enhanced duplicate address detection. **Since: 4.18.1
globalAddressesAreVirtual **	*boolean*	Whether configured addresses are physical or virtual. **Since: 4.24.2
[interfaceAddressBrief] **	*InterfaceAddressBrief* see 2.467.1.3	Network layer address for this interface. **Since: 4.24.2
interfaceStatus	*enum: signal | inactive | connected | unknown | maint-down | errdisabled | disabled | admin | notapplicable | notconnect | noSignal | initializing | uninitialized*	Connection status of the interface.
[ipv4Routable0] **	*boolean*	Whether or not IPv4 routable 0.0.0.0/8is enabled. **Since: 4.29.2
[ipv4Routable240] **	*boolean*	Whether or not IPv4 routable 240.0.0.0/4is enabled. **Since: 4.24.2
lineProtocolStatus	*enum: notPresent | lowerLayerDown | unknown | testing | notApplicable | down | dormant | up*	Line protocol status.
[linkLocal]	*IpAddress* see 2.467.1.5.1	Link-local IPv6 address of this interface.
[maxMssEgress] **	*integer*	TCP MSS ceiling in bytes enforced on packets forwarded to the network. **Since: 4.21.3

[maxMssIngress] **	*integer*	TCP MSS ceiling in bytes enforced on packets arriving from the network. **Since: 4.21.3
mtu **	*integer*	MTU (bytes). **Since: 4.23.1
multicastGroupAddresses	*list of string*	Addresses of multicast groups joined.
name	*interface name*	Name of the interface.
ndReachableTime	*integer*	Neighbor discovery reachable time (in milliseconds).
ndRetransmitInterval	*integer*	Neighbor discovery retransmit interval (in milliseconds).
[neighborDiscoverDetails]	*NeighborDiscoverDetails* see 2.743.1.1	Neighbor discovery Rds.
state	*enum: disabled* | *enabled* | *stalled*	Whether or not the interface is enabled, or stalled in case of a DAD failure on the link-local IP.
[urpf]	*enum: strict* | *loose* | *strictDefault* | *disable* | *looseNonDefault*	Unicast Reverse Path Forwarding mode on the interface.
[urpfV4V6Mismatch]	*boolean*	Whether V4 and V6 urpf conflict on the same interface.
[vrf]	*string*	Name of the VRF.

2.743.1.1 NeighborDiscoverDetails

ndAdvertisedReachableTime	*integer*	Neighbor discovery reachable time being advertised (in milliseconds).
ndRouterAdvInterval	*integer*	Interval in milliseconds between neighbor discovery route advertisements.
ndRouterAdvRetransmit-Interval	*integer*	Neighbor discovery retransmit interval for Router Advertisement (in millisecond).
ndRouterAdvSuppressed	*boolean*	Whether or not Neighbor Discovery Router Advertisement messages are enabled.
routerAdvHopLimit	*integer*	Current hop limit to use in Router Advertisements.
routerAdvLifetime	*integer*	Router lifetime in Router Advertisement messages.
routerPreference	*enum: Medium* | *High* | *Low*	Router preference to be used in Router Advertisements.

2.744 show ipv6 interface brief [...]

interfaces	*table(Interface name) = Ip6StatusBrief* see 2.744.1	Maps an interface name to its abbreviated IPv6 status.

2.744.1 Ip6StatusBrief

[addrSource]	*enum: slaac* | *manual*	How the address is configured on the interface.
addresses	*list of IpAddress* see 2.467.1.5.1	Addresses configured on this interface.
[interfaceAddressBrief] **	*InterfaceAddressBrief* see 2.467.1.3	Network layer address for this interface. **Since: 4.24.2
interfaceStatus	*enum: signal* | *inactive* | *connected* | *unknown* | *maint-down* | *errdisabled* | *disabled* | *admin* | *notapplicable* | *notconnect* | *noSignal* | *initializing* | *uninitialized*	Connection status of the interface.
[ipv4Routable0] **	*boolean*	Whether or not IPv4 routable 0.0.0.0/8is enabled. **Since: 4.29.2
[ipv4Routable240] **	*boolean*	Whether or not IPv4 routable 240.0.0.0/4is enabled. **Since: 4.24.2
lineProtocolStatus	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Line protocol status.
[linkLocal]	*IpAddress* see 2.467.1.5.1	Link-local IPv6 address of this interface.
mtu	*integer*	MTU (bytes).
name	*interface name*	Name of the interface.

2.745 show ipv6 nd proxy [...]

ndProxyInterfaces	*table(Interface name) = NdProxyPrefixInterface* see 2.745.1	A mapping of Interfaces to ND Proxy prefixes.

2.745.1 NdProxyPrefixInterface

ip6Configured	*boolean*	Interface has an IPv6 address configured.
ndProxyConnected	*boolean*	ND Proxy connected prefix is enabled.
ndProxyPrefixes	*list of IpAddress* see 2.467.1.5.1	Prefixes with ND Proxy enabled.

2.746 show ipv6 nd ra internal state [...]

revision 2

interfaces	*table(string) = Interface* see 2.746.1	A mapping of interface ID to interface information.
vrfs	*table(string) = VrfGlobalCounters* see 2.746.2	A mapping of VRF name to global counter information.

2.746.1 Interface

dnsServersLifetime	*integer*	DNS servers lifetime (seconds).
dnsSuffixesLifetime	*integer*	DNS suffixes lifetime (seconds).
hopLimit	*integer*	Router advertisements hop limit value.
initialized	*boolean*	Interface SM is initialized.
intfIndex	*integer*	Interface index.
lastRaTime	*floating point number*	Last router advertisement sent time (epoch).
lastRsRspSentTime	*floating point number*	Last router solicitation response time (epoch).
managedConfigFlag	*boolean*	Managed address configuration flag is set.
mtu	*integer*	Interface status MTU.
nextTimeout	*floating point number*	Next timeout (seconds).
nsRetransmitInterval	*integer*	Time between neighbor solicitation retransmissions.
numDnsServers	*integer*	Number of DNS servers.
numDnsSuffixes	*integer*	Number of DNS domain name suffixes.
numIpv6Addr	*integer*	Number of IPv6 interfaces.
numPktDroppedUnexpectedType	*integer*	Number of packets dropped due to unexpected type.
numPrefixToAdvertise	*integer*	Number of prefixes to advertise.
numPrefixToSuppress	*integer*	Number of prefixes to suppress.
numRaRcvd	*integer*	Number of route advertisements received.
numRaRcvdConsistent	*integer*	Number of consistent route advertisements received.
numRaRcvdInconsistent	*integer*	Number of inconsistent route advertisements received.
numRaSent	*integer*	Number of route advertisements sent.
numRsDroppedInvalidHopLimit	*integer*	Number of router solicitation requests droppped as the hoplimit is less than 255.
numRsRcvd	*integer*	Number of router solicitation requests received.
numRsRspSent	*integer*	Number of router solicitation responses sent.
numRsSuppressed	*integer*	Number of router solicitation requests received with the response suppressed.
otherConfigFlag	*boolean*	Other stateful configuration flag is set.
raIntervalMaxMsec	*integer*	Router advertisements maximum interval (milliseconds).
raIntervalMinMsec	*integer*	Route advertisements minimum interval (milliseconds).

raLifetime	*integer*	Router lifetime value in router advertisements.
raMtuSuppress	*boolean*	Suppress IPv6 RA MTU option is set.
raNotSentIntfNotReady	*boolean*	No router advertisements sent on this interface.
raSuppress	*boolean*	Router advertisements are suppressed.
reachableTime	*integer*	The amount of time that a remote IPv6 node is considered reachable.
routerPreference	*integer*	Router preference type.
rsRspSuppress	*boolean*	Router solicitation responses are suppressed.
vrf	*string*	VRF name.

2.746.2 VrfGlobalCounters

ipv6UnicastRouting	*boolean*	IPv6 unicast routing is enabled.
numCmsgDataErrors	*integer*	Number of Cmessage data errors.
numPktDroppedIntfStateNotFound	*integer*	Number of packets dropped due to interface state not found.
numPktDroppedUnexpectedType	*integer*	Number of packets dropped due to unexpected type.
numRecvMsgErrors	*integer*	Number of received message errors.
numRsDroppedInvalidHopLimit	*integer*	Number of routes dropped due to invalid hop limit.

2.747 show ipv6 nd ra neighbors [...]

vrfs	*table(string) = InterfaceNeighbors* see 2.747.1	A mapping of VRF name to its neighbors.

2.747.1 InterfaceNeighbors

interfaces	*table(string) = RouterNeighbors* see 2.747.1.1	A mapping of interface to its neighbors.

2.747.1.1 RouterNeighbors

routerNeighbors	*list of RouterNeighbor* see 2.747.1.1.1	IPv6 link local address of neighbors discovered byRouter Advertisement.

2.747.1.1.1 RouterNeighbor

address	*IPv6 address*	IPv6 link local address of neighbor.
lastRa	*floating point number*	UTC time at which the last RA was received fromthe neighbor.

2.748 show ipv6 neighbor [...]

revision 2

dynamicEntries	*integer*	Number of dynamically learnt addresses in ARP table.
ipV6Neighbors	*list of IpV6Neighbor* see 2.748.1	IPv6 neighbors discovered by neighbor discovery.
notLearnedEntries	*integer*	Number of addresses in ARP table not due to learning.
staticEntries	*integer*	Number of statically configured addresses in ARP table.
totalEntries	*integer*	Total number of addresses in ARP table.

2.748.1 IpV6Neighbor

address	*string*	IPv6 address or hostname.
[age]	*integer*	How many minutes ago the neighbor entry was learned.
[extra]	*string*	Extra metadata about this neighbor's discovery.
hwAddress	*string*	Hardware (layer 2) address learnt for the corresponding network (layer 3) address.
interface	*string*	Interface name the neighbor was learned on.

2.749 show ipv6 ospf [...]

vrfs	*table(string) = InstList-Ospf3ShowInstModel* see 2.749.1	OSPFv3 instance information for all VRFs.

2.749.1 InstListOspf3ShowInstModel

instList	*table(numeric string) = Ospf3ShowInstModel* see 2.749.1.1	OSPF Instance list keyed by instance id.

2.749.1.1 Ospf3ShowInstModel

abrInstance	*boolean*	This OSPFv3 instance is an Area Border Router.
[adjacencyExchangeStart-Threshold]	*integer*	Adjacency exchange-start threshold.
[areaList]	*table(IPv4 address) = Ospf3ShowInstAreaModel* see 2.749.1.1.1	List of areas in this OSPFv3 instance.
asbrInstance	*boolean*	This OSPFv3 instance is an AS Border Router.
[ecmpMaximumNexthops]	*integer*	Maximum number of ECMP nexthops.
fipsStatus **	*boolean*	Indicates if FIPS mode is successfully enabled. **Since: 4.20.5
floodPacing **	*integer*	Flood packet pacing timer in msecs. **Since: 4.20.0
[gracefulRestartInfo] **	*OspfShowInstGraceful-RestartInfoModel* see 2.642.1.1.4	Graceful restart information. **Since: 4.18.0
[instanceId] **	*integer*	OSPFv3 instance ID. **Since: 4.17.0
[lsaInformation]	*Ospf3ShowInstLsaThrottle-TimersInfoModel* see 2.749.1.1.3	LSA information.
lsaLimitInformation **	*Ospf3ShowInstLsaLimit-InfoModel* see 2.749.1.1.2	LSA limit information. **Since: 4.26.2
[maxMetricTypeInformation]	*Ospf3ShowInstMaxMetric-TypeInfoModel* see 2.749.1.1.4	Max metric type information.
[maxMetricUnsetInformation]	*Ospf3ShowInstMaxMetric-UnsetInfoModel* see 2.749.1.1.5	Max metric unset information.
numAreas	*integer*	Number of areas.
numBackboneNeighbors **	*integer*	Number of backbone neighbors. **Since: 4.20.5
numNeighbors	*integer*	Number of neighbors.
numNormalAreas	*integer*	Number of normal areas.
numNssaAreas	*integer*	Number of nssa areas.
numStubAreas	*integer*	Number of stub areas.

processId	*integer*	OSPFv3 process ID.
[referenceBandwidth] **	*integer*	Auto-cost reference bandwidth in Mbps. **Since: 4.15.3
[routerId]	*IPv4 address*	OSPFv3 Router id.
shutDown	*boolean*	Indicates if OSPFv3 instance is currently shut down.
[spfInformation]	*Ospf3ShowInstSpfThrottle-TimersInfoModel* see 2.749.1.1.6	SPF information.
[vrf]	*string*	VRF name.

2.749.1.1.1 Ospf3ShowInstAreaModel

[areaAuthTypeInformation]	*Ospf3AreaAuthInfoModel* see 2.749.1.1.1.1	Area authentication information.
[areaFilterPrefixList] **	*string*	Area Filter prefix-list. **Since: 4.26.2
areaFiltersConfigured **	*boolean*	Area summary filters configured. **Since: 4.26.0
areaId	*IPv4 address*	OSPFv3 area ID.
areaType	*enum: stubArea | nssaArea | normalArea*	Type of area(normal/stub/nssa).
nssaTransState	*boolean*	Nssa translate always flag for area.
numIntf	*integer*	Number of interfaces in the area.
rangeList	*table(IPv6 address) = Ospf3ShowInstAreaRange-Model* see 2.749.1.1.1.2	List of address ranges in this OSPFv3 area.
spfCount **	*integer*	Area SPF count. **Since: 4.20.5

2.749.1.1.1.1 Ospf3AreaAuthInfoModel

areaAuthConfig	*boolean*	area authenticationis enabled or not.
[authProtocol]	*enum: IPPROTO_ESP | IPPROTO_AH*	OSPFv3 area authentication protocol.
[hmacAlgorithm]	*enum: HMAC-MD5 | HMAC-SHA1*	HMAC algorithm being used.
[secParamIndex]	*integer*	IPsec Security paramater index.

2.749.1.1.1.2 Ospf3ShowInstAreaRangeModel

address	*IPv6 address*	Area range address.
cost	*integer*	Area range cost.
[doNotAdvertise] **	*boolean*	Do not advertise the area range. The cost value is invalid if this flag is set to true. **Since: 4.17.0
maskLen	*integer*	Area range mask length.

2.749.1.1.2 Ospf3ShowInstLsaLimitInfoModel

allowedIncidentCount	*integer*	LSA limit allowed incident count.
clearTimeout	*integer*	LSA limit clear timeout (mins).
currentIncidentCount	*integer*	LSA limit current incident count.
disabled	*boolean*	disabled due to LSA limit.
disabledTime	*integer*	LSA limit disabled time (mins).
[disabledTimeEnd]	*floating point number*	LSA limit disabled time end.
exceedAction	*string*	LSA limit exceed-action option.
lsaLimit	*integer*	Maximum number of LSAs allowed.
warningLimit	*integer*	LSA limit threshold for warning message.

2.749.1.1.3 Ospf3ShowInstLsaThrottleTimersInfoModel

lsaArrivalInterval	*integer*	LSA arrival interval in msecs.
lsaHoldInterval	*integer*	LSA hold interval in msecs.
lsaMaxWaitInterval	*integer*	LSA max wait interval in msecs.
lsaStartInterval	*integer*	LSA start interval in msecs.
numLsa **	*integer*	Number of LSAs. **Since: 4.20.0

2.749.1.1.4 Ospf3ShowInstMaxMetricTypeInfoModel

[maxMetricCfgDuration]	*integer*	Max metric config durationin secs.
maxMetricCondition	*boolean*	Originating router LSA with max metric, is configured or not.
[maxMetricExpiryTimestamp]	*integer*	UTC time stamp of expiry of max metricadvertisement.
[maxMetricExternal]	*integer*	Max metric for external LSA.
maxMetricState	*boolean*	Max metric advertisements being currently active or not.
[maxMetricStub]	*boolean*	Max metric including stubs.
[maxMetricSummary]	*integer*	Max metric for summary LSA.
maxMetricWaitBgp	*boolean*	Max metric while Bgp is converging.

2.749.1.1.5 Ospf3ShowInstMaxMetricUnsetInfoModel

[maxMetricUnsetOrig-Duration]	*integer*	Max metric originate duration (secs).
[maxMetricUnsetReason]	*enum: UserConfig* | *BgpConverged* | *BgpTimerExpired* | *TimerExpired*	Max metric unset reason.
[maxMetricUnsetTimestamp]	*integer*	Max metric unset timestamp.

2.749.1.1.6 Ospf3ShowInstSpfThrottleTimersInfoModel

lastSpf	*integer*	Last SPF run (secs).
nextSpf	*integer*	Next SPF run (msecs).

spfCurrHoldInterval	*integer*	SPF current hold interval in msecs.
spfHoldInterval	*integer*	SPF hold interval in msecs.
spfMaxWaitInterval	*integer*	SPF max wait interval in msecs.
spfStartInterval	*integer*	SPF start interval in msec.

2.750 show ipv6 ospf access-list [...]

see 2.53

2.751 show ipv6 ospf border-routers [...]

vrfs	*table(string) = InstListOspf3Border-RoutersInstance* see 2.751.1	OSPFv3 border routers information for all VRFs.

2.751.1 InstListOspf3BorderRoutersInstance

instList	*table(numeric string) = Ospf3BorderRouters-Instance* see 2.751.1.1	OSPFv3 Instance list keyed by instance id.

2.751.1.1 Ospf3BorderRoutersInstance

ospf3BorderRouterEntries	*table(IPv4 address) = Ospf3BorderRoutersEntry* see 2.751.1.1.1	Dictionary of OSPFv3 border router entries indexed by area-id.
processId	*integer*	OSPFv3 process identifier.
vrf	*string*	VRF name.

2.751.1.1.1 Ospf3BorderRoutersEntry

borderRouters	*table(IPv4 address) = Ospf3BorderRoutersType* see 2.751.1.1.1.1	Dictionary of OSPFv3 border routers type indexed by router-id.

2.751.1.1.1.1 Ospf3BorderRoutersType

abr	*boolean*	Router is an area border router.
asbr	*boolean*	Router is an autonomous system boundary router.

2.752 show ipv6 ospf database [...]

vrfs	*table(string) = InstList-Ospf3DatabaseInstance* see 2.752.1	OSPFv3 database information for all VRFs.

2.752.1 InstListOspf3DatabaseInstance

instList	*table(numeric string) = Ospf3DatabaseInstance* see 2.752.1.1	OSPFv3 Instance list keyed by instance id.

2.752.1.1 Ospf3DatabaseInstance

ospf3AreaEntries	*table(IPv4 address) = Ospf3AreaModel* see 2.752.1.1.1	A mapping between an Area and its LSA list.
ospf3AsLsas	*list of Ospf3Lsa* see 2.752.1.1.1.1	List of LSAs with AS scope.
ospf3InterfaceEntries	*table(Interface name) = Ospf3InterfaceModel* see 2.752.1.1.2	A mapping between an Interface and its LSA list.

2.752.1.1.1 Ospf3AreaModel

ospf3AreaLsaList	*list of Ospf3Lsa* see 2.752.1.1.1.1	List of LSAs for the area.

2.752.1.1.1.1 Ospf3Lsa

advertisingRouter	*IPv4 address*	LSA advertising router.
age	*integer*	LSA age in seconds.
checksum	*integer*	LSA checksum.
[exceptionFlag]	*enum: announcing-MaximumMetric | announcingMaximumCost*	Indicates maximum cost being advertised for self originated LSAs.
length	*integer*	LSA length.
linkStateId	*IPv4 address*	LSA link state id.
lsaType	*enum: graceLsa | interAreaPrefixLsa | routerLsa | asExternalLsa | interAreaRouterLsa | intraAreaPrefixLsa | linkLsa | networkLsa | nssaLsa*	LSA type.
[ospf3ExternalLsa]	*Ospf3ExternalLsa* see 2.752.1.1.1.2	AS external or NSSA LSA.
[ospf3InterAreaPrefixLsa]	*Ospf3InterAreaPrefixLsa* see 2.752.1.1.1.5	Inter area prefix LSA.
[ospf3InterAreaRouterLsa]	*Ospf3InterAreaRouterLsa* see 2.752.1.1.1.6	Intra area router LSA.
[ospf3IntraAreaPrefixLsa]	*Ospf3IntraAreaPrefixLsa* see 2.752.1.1.1.8	Intra area prefix LSA.

[ospf3LinkLsa]	*Ospf3LinkLsa* see 2.752.1.1.1.9	Link LSA.
[ospf3NetworkLsa]	*Ospf3NetworkLsa* see 2.752.1.1.1.10	Network LSA.
[ospf3RouterLsa]	*Ospf3RouterLsa* see 2.752.1.1.1.11	Router LSA.
sequenceNumber	*integer*	LSA sequence number.

2.752.1.1.1.2 Ospf3ExternalLsa

[externalRouteTag]	*integer*	External route tag.
[forwardingAddress]	*IP address*	Forwarding address if available.
metric	*integer*	Route cost.
metricType	*enum: type1 | type2*	External metric type (E bit) type1 : External metric Type 1 type2 : External metric Type 2.
prefix	*Ospf3LsaPrefix* see 2.752.1.1.1.3	Prefix adverstised in the LSA.

2.752.1.1.1.3 Ospf3LsaPrefix

[metric]	*integer*	Prefix cost.
prefix	*IP address*	Prefix advertised.
prefixLength	*integer*	Prefix length.
prefixOptions	*Ospf3LsaPrefixOptions* see 2.752.1.1.1.4	Prefix options.

2.752.1.1.1.4 Ospf3LsaPrefixOptions

doNotUseInRouteCalc **	*boolean*	A type 3,5 or 7 LSA is sent from a PE router to a CE router. **Since: 4.25.0
localAddressCapability	*boolean*	Prefix scope is IPv6 link local.
multicastCapability	*boolean*	Originating router supports multicast extensions to OSPFv3.
noUnicastCapability	*boolean*	Prefix excluded from IPv6 unicast calculations.
propagateNssaPrefix	*boolean*	Prefix propagated by NSSA area border routers.

2.752.1.1.1.5 Ospf3InterAreaPrefixLsa

metric	*integer*	Route cost.
prefix	*Ospf3LsaPrefix* see 2.752.1.1.1.3	Prefix advertised in the LSA.

2.752.1.1.1.6 Ospf3InterAreaRouterLsa

destinationRouterId	*IPv4 address*	Router id of the destination router.
metric	*integer*	Route cost.

options	*Ospf3IntfOptions* see 2.752.1.1.1.7	Optional capabilities supported by the destination router.

2.752.1.1.1.7 Ospf3IntfOptions

activeRouter	*boolean*	Originator is an Active-router.
[addressFamily] **	*boolean*	Router supports address families. **Since: 4.17.0
demandCircuit	*boolean*	Demand Circuit Handling.
ipv6	*boolean*	Include in IPV6 Route Calculations.
multicastForwarding	*boolean*	IP Multicast datagram forwarding.
nonStubArea	*boolean*	Attached to a non-stub area.
nssa	*boolean*	Attached to a Not-so-stubby-area.

2.752.1.1.1.8 Ospf3IntraAreaPrefixLsa

numPrefixes	*integer*	Number of prefixes in intra area prefix LSA.
prefixList	*list of Ospf3LsaPrefix* see 2.752.1.1.1.3	List of prefixes in intra area prefix LSA.
referencedAdvertisingRouter	*IPv4 address*	Referenced advertising router.
referencedLinkStateId	*IPv4 address*	Referenced link state id.
referencedLsaType	*enum: graceLsa | interAreaPrefixLsa | routerLsa | asExternalLsa | interAreaRouterLsa | intraAreaPrefixLsa | linkLsa | networkLsa | nssaLsa*	Referenced LSA type.

2.752.1.1.1.9 Ospf3LinkLsa

linkLocalAddress	*IP address*	Link local address of the originating router.
numPrefixes	*integer*	Number of prefixes in link LSA.
options	*Ospf3IntfOptions* see 2.752.1.1.1.7	Optional capabalities supported by the originating router.
prefixList	*list of Ospf3LsaPrefix* see 2.752.1.1.1.3	List of prefixes in link LSA.
priority	*integer*	Router priority of the originating router.

2.752.1.1.1.10 Ospf3NetworkLsa

attachedRouters	*list of IPv4 address*	List of routers attached to the link.
options	*Ospf3IntfOptions* see 2.752.1.1.1.7	Optional capabilities supported by the router.

2.752.1.1.1.11 Ospf3RouterLsa

options	*Ospf3RouterLsaOptions* see 2.752.1.1.1.13	Optional capabilities supported by the router.

routerLsaLinks	*list of Ospf3RouterLsa-InterfaceEntry* see 2.752.1.1.1.12	List of router links.

2.752.1.1.1.12 Ospf3RouterLsaInterfaceEntry

interfaceId	*IPv4 address*	Local interface id.
interfaceType	*enum: virtualLink | pointToPoint | transitNetwork | reserved*	Router LSA interface type. pointToPoint : Point-to-Point connected interface transitNetwork : Connected to transit network reserved : Reserved virtualLink : Virtual link.
metric	*integer*	Interface cost.
neighborInterfaceId	*IPv4 address*	Interface id of the neighbor router.
neighborRouterId	*IPv4 address*	Router id of the neighbor router.

2.752.1.1.1.13 Ospf3RouterLsaOptions

areaBorderRouter	*boolean*	Area border router.
asBoundaryRouter	*boolean*	Autonomous system boundary router.
nssaTranslation	*boolean*	Translate NSSA LSAs to AS external LSAs.
virtualLinkEndpoint	*boolean*	Endpoint for virtual link(s) in the area.

2.752.1.1.2 Ospf3InterfaceModel

ospf3InterfaceLsaList	*list of Ospf3Lsa* see 2.752.1.1.1.1	List of LSAs for the interface.

2.753 show ipv6 ospf database database-summary [...]

vrfs	*table(string) = InstList-Ospf3DatabaseSummary* see 2.753.1	OSPFv3 database information for all VRFs.

2.753.1 InstListOspf3DatabaseSummary

instList	*table(numeric string) = Ospf3DatabaseSummary* see 2.753.1.1	OSPFv3 Instance list keyed by instance id.

2.753.1.1 Ospf3DatabaseSummary

areaId	*IP address*	Area identifier.
instanceId	*integer*	Instance ID.
numAsex	*integer*	Number of AS-external LSAs.
numGrace	*integer*	Number of Grace LSAs.
numInterPfx	*integer*	Number of Inter Area Prefix LSAs.
numInterRtr	*integer*	Number of Inter Area Router LSAs.

numIntraPfx	*integer*	Number of Intra-Area Prefix LSAs.
numLink	*integer*	Number of Link LSAs.
numNetwork	*integer*	Number of Network LSAs.
numNssa	*integer*	Number of type-7 NSSA LSAs.
numRouter	*integer*	Number of Router LSAs.
vrf	*string*	VRF name.

2.754 show ipv6 ospf interface [...]

vrfs	*table(string)* = *InstListOspf3Interface* see 2.754.1	OSPFv3 instance information for all VRFs.

2.754.1 InstListOspf3Interface

instList	*table(numeric string)* = *Ospf3Interface* see 2.754.1.1	OSPFv3 Instance list keyed by instance id.

2.754.1.1 Ospf3Interface

indices	*table(Interface name)* = *Ospf3InterfaceEntry* see 2.754.1.1.1	Dictionary of OSPFv3 Interfaces indexed by the interface name.

2.754.1.1.1 Ospf3InterfaceEntry

area	*IPv4 address*	Area assigned to the interface.
[authentication]	*Ospf3IntfAuthInfo* see 2.754.1.1.1.1	Interface authentication details.
backupDesignatedRouter	*IPv4 address*	Backup designated router ID.
bfdState	*enum: disabled | default | enabled*	BFD configuration state.
cost	*integer*	Interface metric.
deadInterval	*integer*	Router dead interval in seconds.
designatedRouter	*IPv4 address*	Designated router ID.
helloInterval	*integer*	hello-interval in seconds.
interfaceAddress	*IP address*	Interface ipv6 link-local address.
interfaceType	*enum: p2p | nbma | p2mp | virtual | broadcast*	OSPFv3 interface type. broadcast - Broadcast interface. nbma - Non-Broadcast Multi-Access interface. virtual - Virtual interface. p2p - Point-to-point interface. p2mp - Point-to-multipoint interface.
[interfaceV4Address] **	*IPv4 address*	Interface ipv4 primary address. **Since: 4.18.0
[interfaceV4MaskLen] **	*integer*	Interface ipv4 primary address prefix len. **Since: 4.18.0
neighborCount	*integer*	Neighbor count.
options	*Ospf3IntfOptions* see 2.752.1.1.1.7	Interface optional capabilities.
passive **	*boolean*	Passive state. **Since: 4.18.0
priority	*integer*	Interface priority.
retransmitInterval	*integer*	Retransmit interval in seconds.

state	*enum: p2p | loopback | backupDr | drOther | down | dr | waiting*	OSPFv3 interface state. down - Initial interface state. loopback - Interface is looped back. waiting - Identity determination of designated routers underway. p2p - Interface connects to a point-to-point network or virtual link. dr - Router is the designated router on the attached network. backupDr - Router is the backup designated router on the network. drOther - Interface is neither DR or BDR, and connects to a DR or BDR.
transmitDelay	*integer*	Transmit delay in seconds.
vrf	*string*	VRF name.

2.754.1.1.1.1 Ospf3IntfAuthInfo

algorithm	*enum: HMAC-MD5 | HMAC-SHA1*	Authentication HMAC Algorithm.
protocol	*enum: ESP | AH*	IP Security Protocol. AH - Authentication Header ESP - Encapsulating Security Payload.
spi	*integer*	Security Parameter Index.

2.755 show ipv6 ospf lsa-log [...]

vrfs	*table(string) = InstListOspf3LsaLogInst* see 2.755.1	OSPFv3 lsa-log for the given VRF.

2.755.1 InstListOspf3LsaLogInst

instList	*table(numeric string) = Ospf3LsaLogInst* see 2.755.1.1	OSPFv3 Instance list keyed by instance id.

2.755.1.1 Ospf3LsaLogInst

ospf3LsaLogEntries	*list of Ospf3LsaLogEntry* see 2.755.1.1.1	List of LSA log entries.
[routerId]	*IPv4 address*	OSPFv3 router identifier.

2.755.1.1.1 Ospf3LsaLogEntry

advertisingRouter	*IPv4 address*	LSA advertising router.
linkStateId	*IPv4 address*	LSA link state identifier.
logTime	*floating point number*	LSA log generation time in seconds.
lsaEventType	*enum: outDelay | backoffRestarted | discarded | backoffStarted | backedOff*	LSA event types: backoffStarted - lsa backoff started backedOff - lsa backed off backoffRestarted - lsa backoff restarted outDelay - lsa delayed discarded - lsa discarded.

`lsaHoldInterval`	*integer*	LSA Hold interval in milliseconds.
`lsaType`	*enum: graceLsa* | *interAreaPrefixLsa* | *routerLsa* | *asExternalLsa* | *interAreaRouterLsa* | *intraAreaPrefixLsa* | *linkLsa* | *networkLsa* | *nssaLsa*	LSA type.
`sequenceNumber`	*integer*	LSA sequence number.

2.756 show ipv6 ospf neighbor [...]

vrfs	*table(string) = InstList-Ospf3NeighborsInstance* see 2.756.1	OSPFv3 neighbor instance information for all VRFs.

2.756.1 InstListOspf3NeighborsInstance

instList	*table(numeric string) = Ospf3NeighborsInstance* see 2.756.1.1	OSPFv3 Instance list keyed by instance id.

2.756.1.1 Ospf3NeighborsInstance

ospf3NeighborEntries	*list of Ospf3NeighborsEntry* see 2.756.1.1.1	List of OSPFv3 neighbor entries.

2.756.1.1.1 Ospf3NeighborsEntry

adjacencyState	*enum: attempt | restart | exchStart | 2Ways | loading | down | init | full | exchange*	OSPFv3 neighbor adjacency state. down - No hello received from neighbor attempt - Router attempting to establish connection init - Hello received but two-way communication not established 2-ways - Bi-directional communication established between two routers exch-start - Exchange of link state information between routers and their DR and BDR are started exchange - Routers exchanging database descriptor packets loading - Exchange of link state information occuring full - Routers are fully adjacent with each other restart - Neighbor is restarting.
areaId	*IPv4 address*	OSPFv3 neighbor area identifier.
backupDesignatedRouter	*IPv4 address*	OSPFv3 neighbor ID of backup designated router.
designatedRouter	*IPv4 address*	OSPFv3 neighbor ID of designated router.
[details]	*Ospf3NeighborDetails* see 2.756.1.1.1.1	OSPFv3 neighbor details.
inactivity	*floating point number*	Timestamp of OSPFv3 neighbor inactivity/dead timer.
interfaceName	*interface name*	OSPFv3 neighbor interface name.
options	*Ospf3NeighborOptions* see 2.756.1.1.1.2	OSPFv3 neighbor options.
priority	*integer*	OSPFv3 neighbor priority.
routerId	*string*	OSPFv3 neighbor router identifier.

2.756.1.1.1.1 Ospf3NeighborDetails

bfdRequestSent	*boolean*	OSPFv3 neighbor BFD request is sent.

bfdState	*enum: down | adminDown | up | init*	OSPFv3 neighbor BFD state. adminDown - Session taken down for administrative purposes init - Waiting for session establishment up - Session established successfully down - Session is down or has just been created.
[grExitReason] **	*enum: grExitGracePeriodExpired | grExitNone | grExitNetworkChanged | grExitHelperDisabled | grExitAdjNotFull | grExitGraceLsaFlushed | grExitGracefulRestartInProgress*	OSPFv3 neighbor graceful restart exit reason. **Since: 4.18.0
[grHelperTimer] **	*floating point number*	Timestamp of OSPFv3 neighbor graceful period expiration. **Since: 4.18.0
[grLastRestartTime] **	*floating point number*	Timestamp of the OSPFv3 neighbor last graceful restart time. **Since: 4.18.0
grNumAttempts **	*integer*	OSPFv3 neighbor graceful restart attempts. **Since: 4.18.0
stateTime **	*floating point number*	OSPFv3 neighbor current state time. **Since: 4.30.1

2.756.1.1.1.2 Ospf3NeighborOptions

activeRouter	*boolean*	Originating router is active or not.
[addressFamily] **	*boolean*	Router supports address families. **Since: 4.17.0
demandCircuitsSupport	*boolean*	Originating router supports OSPFv3 over Demand Circuits.
doNotUseInRouteCalc	*boolean*	A type 3,5 or 7 LSA is sent from a PE router to a CE router.
externalRoutingCapability	*boolean*	Originating router is capable of accepting AS External LSAs.
multicastCapability	*boolean*	Originating router supports multicast extensions to OSPFv3.
nssaCapability	*boolean*	Originating router supports Type-7 NSSA-External-LSAs.

2.757 show ipv6 ospf request-list [...]

vrfs	*table(string) = InstListOspf3LsaRequestList* see 2.757.1	OSPFv3 LSA request list information for all VRFs.

2.757.1 InstListOspf3LsaRequestList

instList	*table(numeric string) = Ospf3LsaRequestList* see 2.757.1.1	OSPF Instance list keyed by instance id.

2.757.1.1 Ospf3LsaRequestList

instanceId	*integer*	OSPFv3 instance identifier.
neighbors	*table(IPv4 address) = Ospf3LsaRequestInstance* see 2.757.1.1.1	Dictionary of request keyed by neighbor router ID.

2.757.1.1.1 Ospf3LsaRequestInstance

interfaceAddress	*IP address*	OSPFv3 neighbor interface address.
interfaceName	*interface name*	OSPFv3 neighbor interface name.
requestQueueLength	*integer*	LSA request queue length.
requests	*list of Ospf3LsaEntry* see 2.757.1.1.1.1	LSA request list.

2.757.1.1.1.1 Ospf3LsaEntry

advertisingRouter	*IPv4 address*	LSA advertising router.
age	*integer*	LSA age in seconds.
checksum	*integer*	LSA checksum.
linkStateId	*IPv4 address*	LSA link state identifier.
lsaType	*enum: graceLsa | interAreaPrefixLsa | routerLsa | asExternalLsa | interAreaRouterLsa | intraAreaPrefixLsa | linkLsa | networkLsa | nssaLsa*	LSA type.
sequenceNumber	*integer*	LSA sequence number.

2.758 show ipv6 ospf retransmission-list [...]

vrfs	*table(string) = InstListOspf3Lsa-RetransmissionList* see 2.758.1	OSPFv3 LSA retransmission information for all VRFs.

2.758.1 InstListOspf3LsaRetransmissionList

instList	*table(numeric string) = Ospf3LsaRetransmission-List* see 2.758.1.1	OSPF Instance list keyed by instance id.

2.758.1.1 Ospf3LsaRetransmissionList

neighbors	*table(IPv4 address) = Ospf3LsaRetransmission-Instance* see 2.758.1.1.1	Dictionary of retransmission list keyed by neighbor router ID.

2.758.1.1.1 Ospf3LsaRetransmissionInstance

interfaceAddress	*IP address*	OSPFv3 neighbor interface address.
interfaceName	*interface name*	OSPFv3 neighbor interface name.
nextRetransmissionTime	*floating point number*	Next retransmission time in UTC.
retransmissions	*list of Ospf3LsaEntry* see 2.757.1.1.1.1	LSA retransmission list.

2.759 show ipv6 ospf spf-log [...]

vrfs	*table(string) = InstListOspf3SpfLogInst* see 2.759.1	OSPFv3 spf-log for the given VRF.

2.759.1 InstListOspf3SpfLogInst

instList	*table(numeric string) = Ospf3SpfLogInst* see 2.759.1.1	OSPFv3 Instance list keyed by instance id.

2.759.1.1 Ospf3SpfLogInst

ospf3SpfLogEntries	*list of Ospf3SpfLogEntry* see 2.759.1.1.1	OSPFv3 SPF Log Entries.
[routerId]	*IPv4 address*	OSPFv3 router identifier.

2.759.1.1.1 Ospf3SpfLogEntry

spfDuration	*floating point number*	SPF duration in milliseconds.
spfScheduleReason	*string*	Reason for scheduling SPF.
spfStartTime	*floating point number*	SPF start time in UTC.

2.760 show ipv6 prefix-list NAME detail

see 2.687

2.761 show ipv6 prefix-list NAME summary

see 2.688

2.762 show ipv6 prefix-list [...]

see 2.687

2.763 show ipv6 route (PREFIX | ADDR) longer-prefixes [...]

revision 5

vrfs	*table(string) = Ip6Routes* see 2.763.1	Route info of each VRF.

2.763.1 Ip6Routes

allRoutesProgrammedHardware	*boolean*	True if every route is programmed in hardware.
allRoutesProgrammedKernel	*boolean*	True if every route is programmed in kernel.
defaultRouteState	*enum: notReachable | reachable | notSet*	The state of the default route.
routes	*table(string) = Ip6Route* see 2.763.1.1	List of ipv6 routes keyed by prefix.
routingDisabled	*boolean*	True if routing is disabled.

2.763.1.1 Ip6Route

[cbfVias]	*list of CbfVia* see 2.691.1.1.1	List of Class based forwarding vias.
directlyConnected	*boolean*	True if this route is directly connected though an interface, internal, or is a drop route.
hardwareProgrammed	*boolean*	True if this route is programmed in hardware.
kernelProgrammed	*boolean*	True if this route is programmed in the kernel.
[metric]	*integer*	Used by routing protocol to determine whether one route should be chosen over another. Lower metrics have precedence.
[preference]	*integer*	Administrative distance; lower values have precedence.
routeAction	*enum: drop | forward*	What kind of route it is.
routeLeaked	*boolean*	True if this route is leaked from different VRF.
routeType	*enum: connected | bgp | isisL2 | rip | dropRoute | iBGP | dhcp | static | isis | CBFLeaked | isisL1 | kernel | ospf | martian | dynamicPolicy | bgpAggregate | gribi | ospfSummary | eBGP | nexthopGroupStaticRoute*	The ipv6 route type.
[srTePolicy]	*list of RouteSrTePolicy* see 2.691.1.1.1.2	SR-TE policy.
[srcVrf]	*string*	Name of the VRF route is leaked from.
vias	*list of IpRouteVia* see 2.691.1.1.2	List of vias.

2.764 show ipv6 route [...]

revision 5

see 2.763

2.765 show ipv6 route fec [...]

see 2.693

2.766 show ipv6 route fec summary [...]

see 2.695

2.767 show ipv6 route host [...]

revision 2

vrfs	*table(string) = Ip6RoutesHost* see 2.767.1	A mapping of VRF to its IPv6 host routes.

2.767.1 Ip6RoutesHost

allRoutesProgrammedHardware	*boolean*	Every route is programmed in hardware.
routes	*table(string) = Ip6RouteHost* see 2.767.1.1	IPv6 host routes keyed by prefix.
routingDisabled	*boolean*	Routing is disabled.

2.767.1.1 Ip6RouteHost

hardwareProgrammed	*boolean*	True if this route is programmed in hardware.
routeType	*enum: attached | FIB | broadcast | receive*	The ipv6 route host type.
vias	*list of IpRouteHostVia* see 2.767.1.1.1	List of host vias.

2.767.1.1.1 IpRouteHostVia

[interface]	*interface name*	Name of the interface.
[nexthopAddr]	*IP address*	Nexthop address of the via.
toCpu	*boolean*	True if route goes to cpu.

2.768 show ipv6 route kernel unprogrammed [...]

see 2.697

2.769 show ipv6 route multicast [...]

revision 3

routes	*table(IPv4 or IPv6 address prefix) = MulticastRibRoute* see 2.698.1	A mapping from IPv6 prefix to MulticastRibRoute.

2.770 show ipv6 route next-hop nexthop-group [...]

revision 5

see 2.763

2.771 show ipv6 route rpf unicast [...]

[interfaces]	*table(Interface name) = Ip6UrpfInterface* see 2.771.1	Mapping of interface to the additional interface state details.
[prefix]	*IP address prefix*	Destination Prefix.
[viaDefaultRoute]	*boolean*	uRPF carried out via the default route.

2.771.1 Ip6UrpfInterface

allowDefault	*boolean*	Packets are permitted to come in on the default interface.
[lookupVrf] **	*string*	Name of uRPF lookup VRF. **Since: 4.28.2
uRpfMode	*enum: loose | strictDefault | disable | strict*	URPF mode state.

2.772 show ipv6 route static not-installed bfd [...]

see 2.702

2.773 show ipv6 route summary [...]

revision 3

protoModelStatus **	*ServiceRoutingProtoModelStatus* see 2.703.2	Active and configured protocol models. **Since: 4.23.0
vrfs	*table(string) = Ip6RouteSummaryForVrf* see 2.773.1	Route summary for each VRF.

2.773.1 Ip6RouteSummaryForVrf

aggregate	*integer*	Number of aggregate routes.
attached	*integer*	Number of attached routes.
bgpCounts6	*bgpTotals6* see 2.773.1.1	BGP route count breakdown.
connected	*integer*	Number of connected routes.
[dhcp]	*integer*	Number of dhcp routes.
dynamicPolicy **	*integer*	Number of dynamic policy routes. **Since: 4.21.0
gribi **	*integer*	Number of gRIBI routes. **Since: **currentRelease**
internal	*integer*	Number of internal routes.
isisCounts6	*isisTotals6* see 2.773.1.2	IS-IS route count breakdown.
maskLen	*table(numeric string) = integer*	Counts for each mask length.
ospfCounts6	*ospfTotals6* see 2.773.1.3	OSPF route count breakdown.
static	*integer*	Number of static routes.
staticNexthopGroup	*integer*	Number of static nexthop-group routes.
staticNonPersistent	*integer*	Number of static routes configured using the SDK with persistent=False.
staticPersistent	*integer*	Number of static routes configured to persist.
totalRoutes	*integer*	Total number of routes.
[vcs]	*integer*	Number of vxlan-control-service routes.
[vrfCount]	*integer*	Total number of vrf instances.

2.773.1.1 bgpTotals6

bgpExternal	*integer*	Number of bgp External routes.
bgpInternal	*integer*	Number of bgp Internal routes.
bgpLocal **	*integer*	Number of bgp Other routes. **Since: 4.22.1
bgpTotal	*integer*	Total number of bgp routes.

2.773.1.2 isisTotals6

isisLevel1	*integer*	Number of isis Level-1 routes.
isisLevel2	*integer*	Number of isis Level-2 routes.
isisTotal	*integer*	Total number of isis routes.

2.773.1.3 ospfTotals6

nssaExternal1	*integer*	Number of NSSA External-1 routes.
nssaExternal2	*integer*	Number of NSSA External-2 routes.
ospfExternal1	*integer*	Number of ospf External-1 routes.
ospfExternal2	*integer*	Number of ospf External-2 routes.
ospfInterArea	*integer*	Number of ospf Inter-area routes.
ospfIntraArea	*integer*	Number of ospf Intra-area routes.
ospfTotal	*integer*	Total number of ospf routes.

2.774 show ipv6 route vrf all summary brief [...]

see 2.773.1

2.775 show ipv6 user route [...]

see 2.716

2.776 show ipv6 virtual-router [...]

revision 2

see 2.720

2.777 show isis (area proxy) | (INSTANCE area proxy) | (area proxy vrf VRFNAME)

vrfs	*table(string) = Instances-IsisShowAreaProxyModel* see 2.777.1	Area proxy for all VRFs.

2.777.1 InstancesIsisShowAreaProxyModel

isisInstances	*table(string) = IsisShowAreaProxyModel* see 2.777.1.1	IS-IS Instance list keyed by instance name.

2.777.1.1 IsisShowAreaProxyModel

`entry`	*IsisShowAreaProxyEntryModel* see 2.777.1.1.1	Show area proxy information.

2.777.1.1.1 IsisShowAreaProxyEntryModel

`proxyHostname`	*string*	Area proxy hostname.
`proxySysId`	*string*	Area proxy system identifier.
`status`	*string*	Area proxy status.

2.778 show isis (database (detail | traffic-engineering)) | (INSTANCE database (detail | traffic-engineering)) | (database (detail | traffic-engineering) vrf VRFNAME) [...]

vrfs	*table(string) = Instances-IsisLsdbDetailModel* see 2.778.1	IS-IS instance information for all VRFs.

2.778.1 InstancesIsisLsdbDetailModel

isisInstances	*table(string) = IsisLsdbDetailModel* see 2.778.1.1	IS-IS Instance list keyed by instance name.

2.778.1.1 IsisLsdbDetailModel

level	*table(numeric string) = LevelDetailModel* see 2.778.1.1.1	A mapping of isis level to LSDB.

2.778.1.1.1 LevelDetailModel

lsps	*table(string) = LspDetailModel* see 2.778.1.1.1.1	A mapping of systemId to LSP in LSDB.

2.778.1.1.1.1 LspDetailModel

[addrFamily]	*string*	Address Family.
[appSrlgInfos] **	*list of ApplicationSrlg* see 2.778.1.1.1.2	Application Specific Shared Risk Link Group information. **Since: 4.26.0
areaAddresses	*list of LspAreaAddress* see 2.778.1.1.1.8	List of Area Addresses.
[areaProxySubTlvs] **	*list of AreaProxySubTlv* see 2.778.1.1.1.5	Area proxy information. **Since: 4.24.2
[auth]	*LspAuthModel* see 2.778.1.1.1.9	Auth details for the LSP.
checksum	*integer*	Checksum value.
[dynFloodNodeId] **	*list of LspDynFloodNodeId* see 2.778.1.1.1.10	Dynamic Flooding Area Node IDs. **Since: 4.23.0
[dynFloodPath] **	*list of LspDynFloodPath* see 2.778.1.1.1.11	Dynamic Flooding Paths. **Since: 4.23.0
expiryTime	*floating point number*	UTC time for Future Expiry time of LSP.
[flags]	*LspFlags* see 2.778.1.1.1.12	IS-IS Lsp flags.
[hostname]	*LspHostname* see 2.778.1.1.1.13	IS-IS hostname.
[interfaceAddresses]	*list of InterfaceAddresses* see 2.778.1.1.1.7	List of Interface Addresses.
[intermediateSystemType]	*enum: L2 | none | L1*	IS-type for LSP.

[ip6SrlgInfos] **	*list of TeIpv6Srlg* see 2.778.1.1.1.53	Traffic Engineering IPv6 Shared Risk Link Group information. **Since: 4.22.0
length **	*integer*	LSP length. **Since: 4.27.1
[lspGenerationTime]	*floating point number*	Next LSP generation time in UTC.
[modifiedLifetime] **	*integer*	Modified remaining lifetime of an LSP (in seconds). **Since: 4.21.3
[multiTopologies]	*list of LspMtModel* see 2.778.1.1.1.14	Multi Topology details for the LSP.
[neighbors]	*list of LspNeighbors* see 2.778.1.1.1.15	List of LSP neighbors Details.
[purgeOriginator] **	*string*	Purge originator. **Since: 4.23.1
[purgeUpstreamNgb] **	*string*	Purge originator upstream neighbor. **Since: 4.23.1
[rcvdLifetime] **	*integer*	Received remaining lifetime of an LSP (in seconds). **Since: 4.21.3
[reachabilities]	*list of LspReachabilities* see 2.778.1.1.1.23	Reachabilities Information of LSP.
[refreshTime] **	*floating point number*	Future refresh time of LSP in UTC. **Since: 4.21.3
[routerCapabilities]	*list of RouterCapability* see 2.778.1.1.1.27	Router Capability Information.
sequence	*integer*	Sequence number for this LSP.
[srBindings]	*list of SrBindingModel* see 2.778.1.1.1.46	Segment Routing Binding Information.
[srlgInfos]	*list of TeSrlg* see 2.778.1.1.1.56	Traffic Engineering Shared Risk Link Group information.
[srv6Locators] **	*list of Srv6Locators* see 2.778.1.1.1.48	SRv6 Locators information of LSP. **Since: 4.29.0
[supportedInstTlvs] **	*list of SupportedInstTlv* see 2.778.1.1.1.52	Supported instance TLV information. **Since: 4.26.0
[teRouterID]	*TeRouterId* see 2.778.1.1.1.55	IS-IS TE Router-ID.
[unsupportedTlvs] **	*list of UnsupportedTlv* see 2.778.1.1.1.57	Unsupported IS-IS TLV Information. **Since: 4.22.0

2.778.1.1.1.2 ApplicationSrlg

applications	*ApplicationIdentifiers* see 2.778.1.1.1.3	Application identifiers.
[interfaceAddress]	*IP address*	Interface address.
[localLinkId]	*integer*	Local link identifier.
[neighborAddress]	*IP address*	Neighbor address.
ngbSystemId	*string*	System identifier of the neighbor.
[remoteLinkId]	*integer*	Remote link identifier.
srlgIds	*list of SrlgIds* see 2.778.1.1.1.4	List of Shared Risk Link Group IDs.
[v6InterfaceAddress]	*IPv6 address*	IPv6 interface address.

[v6NeighborAddress]	*IPv6 address*	IPv6 neighbor address.

2.778.1.1.1.3 ApplicationIdentifiers

[standardApps]	*list of string*	Standard applications.
[standardAppsUseLegacy]	*boolean*	Use legacy link attributes.
[userDefinedApps]	*list of string*	User defined applications.

2.778.1.1.1.4 SrlgIds

gid	*integer*	Shared Risk Link Group Identifier.

2.778.1.1.1.5 AreaProxySubTlv

[areaProxySystemId]	*string*	Area proxy system ID.
[flags]	*AreaSegmentFlags* see 2.778.1.1.1.6	Area segment flags.
[index]	*integer*	Area segment index/label.
[maskLength] **	*integer*	Area segment IP Prefix Mask Length. **Since: 4.25.1
[prefix] **	*IP address*	Area segment IP Prefix. **Since: 4.25.1

2.778.1.1.1.6 AreaSegmentFlags

areaSegAf	*boolean*	Area segment address family flag.
areaSegLocal	*boolean*	Area segment local flag.
areaSegValue	*boolean*	Area segment value flag.

2.778.1.1.1.7 InterfaceAddresses

[ipv4Address]	*IPv4 address*	IPv4 address.
[ipv6Address]	*IPv6 address*	IPv6 address.

2.778.1.1.1.8 LspAreaAddress

address	*string*	Area Address for lsp.

2.778.1.1.1.9 LspAuthModel

[authKeyId] **	*integer*	Authentication Key Id. **Since: 4.21.3
authLength	*integer*	Authentication Key Length.
authMode	*enum: MD5 | SHA | Text*	Authentication Mode.

2.778.1.1.1.10 LspDynFloodNodeId

endIndex	*integer*	End index of enclosed systems.
startIndex	*integer*	Start index of enclosed systems.

2.778.1.1.1.11 LspDynFloodPath

path	*string*	Path of node indices.

2.778.1.1.1.12 LspFlags

dbOverload	*boolean*	LSP Database Overload.
defaultAttached	*boolean*	Attached to other areas by the Default Metric.
delayAttached	*boolean*	Attached to other areas by the Delay Metric.
errorAttached	*boolean*	Attached to other areas by the Error Metric.
expenseAttached	*boolean*	Attached to other areas by the Expense Metric.
partitionSupport	*boolean*	Support to Partition Repair.

2.778.1.1.1.13 LspHostname

name	*string*	LSP Hostname.

2.778.1.1.1.14 LspMtModel

mtAttached	*boolean*	MultiTopology Attached Flag.
mtOverloaded	*boolean*	MultiTopology Overload Flag.
topologyId	*integer*	Topology Id.

2.778.1.1.1.15 LspNeighbors

TEInfo	*LinkTeInfo* see 2.778.1.1.1.16	TE Information forthe Link.
adjInterfaceAddresses	*list of LspAdjInterfaceAddress* see 2.778.1.1.1.18	List of interface addresses.
adjLanSids	*list of LspAdjLanSid* see 2.778.1.1.1.19	List of LAN Adj SID.
adjNarrowMetricsEnabled	*boolean*	Neighbor narrow metric.
adjSids	*list of LspAdjSid* see 2.778.1.1.1.21	List of Adj SID.
adjTopoId	*integer*	Neighbor topology Id.
[applicationLinkAttributes] **	*list of LspApplicationLinkAttributes* see 2.778.1.1.1.22	List of application specific link attributes. **Since: 4.26.0

[ip6GlobalInterfaceAddress] **	*IPv6 address*	IPv6 global interface address. **Since: 4.22.0
[ip6NeighborAddress] **	*IPv6 address*	Neighbor IPv6 global address. **Since: 4.22.0
metric	*integer*	Link Metric.
neighborAddr	*IPv4 address*	Neighbor Interface address.
systemId	*string*	IS-IS System Id.

2.778.1.1.1.16 LinkTeInfo

[administrativeGroup] **	*integer*	Administrative Group of the Link. **Since: 4.21.3
[maxDelay] **	*integer*	Max unidirectional link delay (us). **Since: 4.26.0
[maxLinkBw]	*integer*	Maximum bandwidth (bps) that can be used on Directed Link.
[maxReservableBw]	*integer*	Maximum bandwidth (bps) that can be reserved on Directed Link.
[maxReservablePriorityBw]	*TeReservablePriority-Bandwidth* see 2.778.1.1.1.17	Maximum bandwidth reservable for a priority.
[metric]	*integer*	TE Link cost (if configured on link).
[minDelay] **	*integer*	Min unidirectional link delay (us). **Since: 4.26.0
[minMaxDelayAnomalous] **	*boolean*	Anomalous min/max link delay. **Since: 4.26.0

2.778.1.1.1.17 TeReservablePriorityBandwidth

priority0	*integer*	Reservable Bandwidth (bps) for priority level 0.
priority1	*integer*	Reservable Bandwidth (bps) for priority level 1.
priority2	*integer*	Reservable Bandwidth (bps) for priority level 2.
priority3	*integer*	Reservable Bandwidth (bps) for priority level 3.
priority4	*integer*	Reservable Bandwidth (bps) for priority level 4.
priority5	*integer*	Reservable Bandwidth (bps) for priority level 5.
priority6	*integer*	Reservable Bandwidth (bps) for priority level 6.
priority7	*integer*	Reservable Bandwidth (bps) for priority level 7.

2.778.1.1.1.18 LspAdjInterfaceAddress

adjInterfaceAddress	*IPv4 address*	Interface Address.

2.778.1.1.1.19 LspAdjLanSid

adjLanFlags	*AdjFlags* see 2.778.1.1.1.20	Adjacency LAN Flags.
adjLanSid	*integer*	Adjacency LAN SID.
adjLanSystemId	*string*	Adjacency SystemId.
adjLanWeight	*integer*	Adjacency LAN Weight.

2.778.1.1.1.20 AdjFlags

adjAf	*boolean*	Sr Adj AddressFamily Flag.
adjBackup	*boolean*	Sr Adj Backup Flag.
adjLocal	*boolean*	Sr Adj Local Flag.
adjSet	*boolean*	Sr Adj Set Flag.
adjValue	*boolean*	Sr Adj Value Flag.

2.778.1.1.1.21 LspAdjSid

adjFlags	*AdjFlags* see 2.778.1.1.1.20	Adjacency Flags.
adjSid	*integer*	Adjacency SID.
adjWeight	*integer*	Adjacency Weight.

2.778.1.1.1.22 LspApplicationLinkAttributes

applications	*ApplicationIdentifiers* see 2.778.1.1.1.3	Application identifiers.
[linkAttributes]	*LinkTeInfo* see 2.778.1.1.1.16	TE link attributes.

2.778.1.1.1.23 LspReachabilities

maskLength	*integer*	Mask Length.
metric	*integer*	Cost of the Adjacency.
metricType	*enum: External | Internal*	Metric Type.
reachabilityNarrowMetrics	*boolean*	Unsupported Reachability Narrow Metrics.
reachabilityTopoId	*integer*	Reachable Topology Id.
reachabilityUpDown	*boolean*	Reachability Status.
[reachabilityV4Addr]	*IPv4 address*	Reachability IPv4 address.
[reachabilityV6Addr]	*IPv6 address*	Reachability IPv6 address.
[routeTags] **	*list of RouteTag* see 2.778.1.1.1.24	List of Route Tags. **Since: 4.21.3
srPrefixReachabilities	*list of SrPrefixReachability* see 2.778.1.1.1.25	List of SR Prefix Reachability.

2.778.1.1.1.24 RouteTag

tag	*integer*	Administrative Tag for IP Prefix.

2.778.1.1.1.25 SrPrefixReachability

algo	*string*	SR Algo Used.
algoNum **	*integer*	SR Algorithm. **Since: 4.25.0
options	*SrPrefixOptions* see 2.778.1.1.1.26	SR Flags for Prefix options.
sid	*integer*	SR Prefix SID.

2.778.1.1.1.26 SrPrefixOptions

explicitNull	*boolean*	SR Explicit Null Flag.
local	*boolean*	Value/Index carried by the Prefix-SID has local significance.
noPenultimate	*boolean*	SR No Penultimate Hop Flag.
nodeSID	*boolean*	SR Node SID Flag.
readvert	*boolean*	SR readvertisement Flag.
value	*boolean*	Prefix-SID carries a value instead of an index.

2.778.1.1.1.27 RouterCapability

[areaLeader] **	*AreaLeaderModel* see 2.778.1.1.1.28	Area Leader. **Since: 4.23.0
[dynamicFlooding] **	*DynamicFloodingModel* see 2.778.1.1.1.29	Dynamic Flooding. **Since: 4.23.0
flags	*RouterCapabilityFlags* see 2.778.1.1.1.42	Router Capability Flags.
flexAlgoDefs **	*list of FlexAlgoDefModel* see 2.778.1.1.1.30	Flexible Algorithm Definitions. **Since: 4.26.0
[msd] **	*NodeMsdModel* see 2.778.1.1.1.41	Node maximum SID depth. **Since: 4.28.0
routerId	*string*	Router Id.
srAlgos	*list of SrAlgo* see 2.778.1.1.1.43	Algorithm used.
srCapabilities	*list of SrCapabilityModel* see 2.778.1.1.1.36	SR Capabilities.
[srlb] **	*SrlbModel* see 2.778.1.1.1.39	SR Local Block Information. **Since: 4.21.3
[srv6Capabilities] **	*Srv6CapabilityModel* see 2.778.1.1.1.44	SRv6 capabilities. **Since: 4.29.0

2.778.1.1.1.28 AreaLeaderModel

algo	*integer*	Area leader algorithm.
prio	*integer*	Area leader priority.

2.778.1.1.1.29 DynamicFloodingModel

enabled	*boolean*	Dynamic flooding enabled.

2.778.1.1.1.30 FlexAlgoDefModel

algorithm	*integer*	Algorithm identifier.
calcType	*enum: SSPF | SPF | Undefined*	Calculation type.
calcTypeValue	*integer*	Calculation type value.
excludeGroups	*list of FlexAlgoExcludeModel* see 2.778.1.1.1.31	Flexible Algorithm Exclude Admin Group.
excludeSrlgGroups	*list of FlexAlgoExcludeSrlgModel* see 2.778.1.1.1.32	Flexible Algorithm Exclude Shared Risk Link Group.
flags	*list of FlexAlgoFlagsModel* see 2.778.1.1.1.33	Flexible Algorithm Flags.
includeAllGroups	*list of FlexAlgoIncludeAllModel* see 2.778.1.1.1.34	Flexible Algorithm Include All Admin Group.
includeAnyGroups	*list of FlexAlgoIncludeAnyModel* see 2.778.1.1.1.35	Flexible Algorithm Include Any Admin Group.
metricType	*enum: TE Metric | Min Unidirectional Delay Metric | IGP Metric | Undefined*	Metric type.
metricTypeValue	*integer*	Metric type value.
prio	*integer*	Priority.

2.778.1.1.1.31 FlexAlgoExcludeModel

group	*string*	Administrative Groups.

2.778.1.1.1.32 FlexAlgoExcludeSrlgModel

group	*integer*	Shared Risk Link Group.

2.778.1.1.1.33 FlexAlgoFlagsModel

flags	*string*	Flags.

2.778.1.1.1.34 FlexAlgoIncludeAllModel

group	*string*	Administrative Groups.

2.778.1.1.1.35 FlexAlgoIncludeAnyModel

group	*string*	Administrative Groups.

2.778.1.1.1.36 SrCapabilityModel

flags	*SrCapabilityFlags* see 2.778.1.1.1.37	SR Capability Flags.
srCapabilitySrgb	*list of SrCapabilitySrgb* see 2.778.1.1.1.38	Segment Routing Global Block Capability Info.

2.778.1.1.1.37 SrCapabilityFlags

mplsV4	*boolean*	Sr Capability mpls V4 Flag.
mplsV6	*boolean*	Sr capability mpls V6 Flag.
srcapV6	*boolean*	Sr Capability V6 Flag.

2.778.1.1.1.38 SrCapabilitySrgb

srgbBase	*integer*	Srgb base.
srgbRange	*integer*	Srgb range.

2.778.1.1.1.39 SrlbModel

srlbRanges	*list of SrlbRange* see 2.778.1.1.1.40	Ranges of labels making up SR Local Block.

2.778.1.1.1.40 SrlbRange

srlbBase	*integer*	Minimum label in range.
srlbRange	*integer*	Length of label range.

2.778.1.1.1.41 NodeMsdModel

[baseMplsImposition]	*integer*	Base MPLS imposition.

2.778.1.1.1.42 RouterCapabilityFlags

rtrCapD	*boolean*	Router Capability D Flag.
rtrCapS	*boolean*	Router Capability S Flag.

2.778.1.1.1.43 SrAlgo

algoNum **	*integer*	SR Algorithm. **Since: 4.25.0
srAlgo	*string*	SR Algo Used.

2.778.1.1.1.44 Srv6CapabilityModel

flags	*Srv6CapabilityFlags* see 2.778.1.1.1.45	SRv6 Capability flags.

2.778.1.1.1.45 Srv6CapabilityFlags

oamSupport	*boolean*	SRv6 OAM capability flag.

2.778.1.1.1.46 SrBindingModel

flags	*SrBindingFlags* see 2.778.1.1.1.47	SR Binding Flags.
label	*integer*	SR Binding Label.
maskLength	*integer*	SR Binding Mask Length.
prefix	*IP address*	SR Binding Ip Prefix.
rangeValue	*integer*	SR Binding Range.
srBindingTopologyId	*integer*	Sr Binding Topology Id.
srPrefixReachabilities	*list of SrPrefixReachability* see 2.778.1.1.1.25	List of SR Reachable Prefixes.
weight	*integer*	SR Binding Weight of the path.

2.778.1.1.1.47 SrBindingFlags

addrFamily	*boolean*	Sr Binding Address Family Flag.
attach	*boolean*	Sr Binding Attach Flags.
flood	*boolean*	Sr Binding flood Flag.
leak	*boolean*	Sr Binding leak Flag.
mirror	*boolean*	Sr Binding Mirror Flag.

2.778.1.1.1.48 Srv6Locators

algorithm	*integer*	Locator algorithm.
[endSids] **	*list of Srv6EndSid* see 2.778.1.1.1.49	End SID information. **Since: 4.30.0
flags	*Srv6LocatorFlags* see 2.778.1.1.1.51	Locator flags.
locatorPrefix	*IP address prefix*	Locator prefix.
locatorTopoId	*integer*	Locator Topology Id.
metric	*integer*	Cost of the Locator reachability.

2.778.1.1.1.49 Srv6EndSid

flavors	*Srv6SidFlavorFlags* see 2.17.1.1.2.30	SID flavors.
[sidStructure]	*Srv6SidStructure* see 2.778.1.1.1.50	SID structure.

sidValue	*IPv6 address*	End SID value.

2.778.1.1.1.50 Srv6SidStructure

argumentLength	*integer*	Argument length in bits.
blockLength	*integer*	Block length in bits.
functionLength	*integer*	Function length in bits.
nodeLength	*integer*	Node length in bits.

2.778.1.1.1.51 Srv6LocatorFlags

down	*boolean*	Up/Down bit.

2.778.1.1.1.52 SupportedInstTlv

instanceId	*integer*	IS-IS Instance Id.
supportedInstances	*list of integer*	List of supported instances.

2.778.1.1.1.53 TeIpv6Srlg

flags	*v6SrlgFlags* see 2.778.1.1.1.54	IPv6 SRLG flags.
ngbSystemId	*string*	System identifier of the neighbor.
srlgIds	*list of SrlgIds* see 2.778.1.1.1.4	List of Shared Risk Link Group IDs.
v6InterfaceAddress	*IPv6 address*	IPv6 interface address.
[v6NeighborAddress]	*IPv6 address*	IPv6 neighbor address.

2.778.1.1.1.54 v6SrlgFlags

neighborAddressIncluded	*boolean*	IPv6 neighbor address is included (NA bit).

2.778.1.1.1.55 TeRouterId

v4RouterId	*IPv4 address*	TE IPv4 Router-Id.
[v6RouterId] **	*IPv6 address*	TE IPv6 Router-Id. **Since: 4.21.3

2.778.1.1.1.56 TeSrlg

flags	*enum: Unnumbered | Numbered*	SRLG flags.
intfAddr	*IP address*	Interface address.
ngbAddr	*IP address*	Neighbor address.
ngbSystemId	*string*	System identifier of the neighbor.

srlgIds	*list of SrlgIds* see 2.778.1.1.1.4	List of Shared Risk Link Group IDs.

2.778.1.1.1.57 UnsupportedTlv

tlvLength	*integer*	TLV length.
tlvType	*integer*	TLV type.

2.779 show isis (dynamic flooding interfaces) | (INSTANCE dynamic flooding interfaces) | (dynamic flooding interfaces vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisDynFloodInterfacesModel* see 2.779.1	Dynamic flooding interfaces for all VRFs.

2.779.1 InstancesIsisDynFloodInterfacesModel

isisInstances	*table(string) = IsisDynFloodInterfacesModel* see 2.779.1.1	IS-IS Instance list keyed by instance name.

2.779.1.1 IsisDynFloodInterfacesModel

levels	*table(numeric string) = IsisDynFloodInterfacesLevelModel* see 2.779.1.1.1	A dict of lists of IS-IS flooding interfaces keyed by level.

2.779.1.1.1 IsisDynFloodInterfacesLevelModel

interfaces	*list of IsisDynFloodInterfacesEntryModel* see 2.779.1.1.1.1	A list of IS-IS flooding paths.

2.779.1.1.1.1 IsisDynFloodInterfacesEntryModel

intf	*interface name*	Interface name.
tempIntf	*boolean*	Temporary interface addition to the flooding topology.

2.780 show isis (dynamic flooding nodes) | (INSTANCE dynamic flooding nodes) | (dynamic flooding nodes vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisDynFloodNodeModel* see 2.780.1	Dynamic flooding nodes for all VRFs.

2.780.1 InstancesIsisDynFloodNodeModel

isisInstances	*table(string) = IsisDynFloodNodeModel* see 2.780.1.1	IS-IS Instance list keyed by instance name.

2.780.1.1 IsisDynFloodNodeModel

levels	*table(numeric string) = IsisDynFloodNodeLevelModel* see 2.780.1.1.1	A dict of IS-IS nodes keyed by level.

2.780.1.1.1 IsisDynFloodNodeLevelModel

nodes	*table(numeric string) = IsisDynFloodNodeEntryModel* see 2.780.1.1.1.1	A dict of IS-IS nodes keyed by index.

2.780.1.1.1.1 IsisDynFloodNodeEntryModel

nodeId	*string*	Node ID.

2.781 show isis (dynamic flooding paths) | (INSTANCE dynamic flooding paths) | (dynamic flooding paths vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisDynFloodPathsModel* see 2.781.1	Dynamic flooding paths for all VRFs.

2.781.1 InstancesIsisDynFloodPathsModel

isisInstances	*table(string) = IsisDynFloodPathsModel* see 2.781.1.1	IS-IS Instance list keyed by instance name.

2.781.1.1 IsisDynFloodPathsModel

levels	*table(numeric string) = IsisDynFloodPathsLevelModel* see 2.781.1.1.1	A dict of lists of IS-IS paths keyed by level.

2.781.1.1.1 IsisDynFloodPathsLevelModel

paths	*list of IsisDynFloodPathsEntryModel* see 2.781.1.1.1.1	A list of IS-IS paths.

2.781.1.1.1.1 IsisDynFloodPathsEntryModel

path	*string*	Path.

2.782 show isis (dynamic flooding topology) | (INSTANCE dynamic flooding topology) | (dynamic flooding topology vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisDynFloodTopologyModel* see 2.782.1	Dynamic flooding topology for all VRFs.

2.782.1 InstancesIsisDynFloodTopologyModel

isisInstances	*table(string) = IsisDynFloodTopologyModel* see 2.782.1.1	IS-IS Instance list keyed by instance name.

2.782.1.1 IsisDynFloodTopologyModel

levels	*table(numeric string) = IsisDynFloodTopologyLevelModel* see 2.782.1.1.1	A dict of lists of IS-IS flooding topology keyed by level.

2.782.1.1.1 IsisDynFloodTopologyLevelModel

topology	*list of IsisDynFloodTopologyPathModel* see 2.782.1.1.1.1	A list of IS-IS flooding paths.

2.782.1.1.1.1 IsisDynFloodTopologyPathModel

index	*integer*	Path index.
path	*list of IsisDynFloodTopologyEntryModel* see 2.782.1.1.1.2	A path in the flooding topology.

2.782.1.1.1.2 IsisDynFloodTopologyEntryModel

name	*string*	A node in a path in the flooding topology.

2.783 show isis (flex-algo) | (INSTANCE flex-algo) | (flex-algo vrf VRF-NAME) [...]

vrfs	*table(string) = InstancesIsisShowFlexAlgoModel* see 2.783.1	Flex-algo for all VRFs.

2.783.1 InstancesIsisShowFlexAlgoModel

isisInstances	*table(string) = IsisShowFlexAlgoModel* see 2.783.1.1	IS-IS Instance list keyed by instance name.

2.783.1.1 IsisShowFlexAlgoModel

details	*list of IsisShowFlexAlgoDetailEntryModel* see 2.783.1.1.1	Flex-algo algorithm details.
routers	*list of IsisShowFlexAlgoRouterModel* see 2.783.1.1.2	Flex-algo participating router.
summaries	*list of IsisShowFlexAlgoSummaryEntryModel* see 2.783.1.1.3	Flex-algo algorithm summaries.

2.783.1.1.1 IsisShowFlexAlgoDetailEntryModel

advertised	*boolean*	Definition advertised.
algorithm	*string*	Algorithm identifier.
alternates	*list of IsisShowFlexAlgoAlternate* see 2.783.1.1.1.1	Other definition advertisers.
color	*integer*	Color for resulting paths.
exclude	*string*	List of administrative groups a link must exclude.
excludeSrlg	*string*	List of shared risk link groups that must be excluded.
includeAll	*string*	List of administrative groups a link must include.
includeAny	*string*	List of administrative groups a link may include.
level	*string*	IS-IS Level.
metric	*enum: igpMetric | minDelay | te*	Metric type.
priority	*integer*	Priority of the definition.
system	*string*	System name.

2.783.1.1.1.1 IsisShowFlexAlgoAlternate

name	*string*	System name.

2.783.1.1.2 IsisShowFlexAlgoRouterModel

algorithm	*string*	Algorithm identifier.
level	*enum: L2 | L1L2 | L1*	IS-IS Level.
routers	*list of IsisShowFlexAlgoRouterEntry* see 2.783.1.1.2.1	Flex-algo algorithm router entry.

2.783.1.1.2.1 IsisShowFlexAlgoRouterEntry

advertising	*boolean*	Router advertising an algorithm definition for this algorithm.
[priority]	*integer*	Priority of the advertised algorithm.
router	*string*	Router name or system id.

2.783.1.1.3 IsisShowFlexAlgoSummaryEntryModel

advertised	*boolean*	Definition advertised.
algorithm	*string*	Algorithm identifier.
level	*string*	IS-IS Level.
metric	*enum: igpMetric | minDelay | te*	Metric type.
system	*string*	System name.

2.784 show isis (graceful-restart) | (INSTANCE graceful-restart) | (graceful-restart vrf VRFNAME)

vrfs	*table(string) = Instances-IsisGracefulRestartModel* see 2.784.1	Graceful Restart information for all VRFs.

2.784.1 InstancesIsisGracefulRestartModel

isisInstances	*table(string) = IsisGracefulRestartModel* see 2.784.1.1	IS-IS Instance list keyed by instance name.

2.784.1.1 IsisGracefulRestartModel

grNeighbors	*table(string) = IsisGracefulRestartNgbEntryModel* see 2.784.1.1.1	A Dict of GracefulRestart Neighbors entries keyed by systemID.
[grState] **	*enum: cancelled | expired*	Graceful Restart State. **Since: 4.27.1
gracefulRestart	*enum: disabled | enabled*	Graceful Restart.
gracefulRestartHelper	*enum: disabled | enabled*	Graceful Restart Helper.
state	*enum: | Restart in progress | Graceful Restart disabled during startup | Last Restart exited after T2 (level-1) expiry | Last Start exited after T2 (level-1) expiry | Last Restart was completed successfully | Last Restart exited after T2 (level-2) expiry | Last Start was completed successfully | Last Start exited after T2 (level-2) expiry | Start in progress | Last Restart exited after T3 expiry*	Graceful Restart State.
systemId	*string*	SystemId.
t1Timer	*integer*	T1 timer - Time after which an unacknowledged start or restart attempt will be repeated (in seconds).
t2ConfigLevel1	*integer*	T2 timer - Maximum time to wait for LSP database synchronization for level-1 (in seconds).
t2ConfigLevel2	*integer*	T2 timer - Maximum time to wait for LSP database synchronization for level-2 (in seconds).
[t2L1Stopped] **	*floating point number*	T2 timer - time when Level-1 stopped. **Since: 4.27.1
[t2L2Stopped] **	*floating point number*	T2 timer - time when Level-2 stopped. **Since: 4.27.1
[t2Level1]	*integer*	Remaining T2 for level-1 (in seconds).

[t2Level2]	*integer*	Remaining T2 for level-2 (in seconds).
[t3Stopped] **	*floating point number*	T3 timer - time when stopped. **Since: 4.27.1
[t3Timer]	*integer*	T3 timer - Time after which the router will declare that it has failed to achieve database synchronization during restart (in seconds).

2.784.1.1.1 IsisGracefulRestartNgbEntryModel

adjacencies	*list of IsisGrNgbListModel* see 2.784.1.1.1.1	List of adjacency data for a neighbor.
[hostname]	*string*	Hostname of Neighbor.

2.784.1.1.1.1 IsisGrNgbListModel

gracefulRestartCapable	*boolean*	Graceful-Restart Capable.
interface	*interface name*	IS-IS interface name.
level	*enum: L2 | L1L2 | L1*	IS-IS level.
status	*enum: running | reStarting | starting | na*	Graceful-Restart State.

2.785 show isis (local-convergence-delay) | (INSTANCE local-convergence-delay) | (local-convergence-delay vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisUloopModel* see 2.785.1	Micro-loop information for all VRFs.

2.785.1 InstancesIsisUloopModel

isisInstances	*table(string) = IsisUloopModel* see 2.785.1.1	IS-IS Instance list keyed by instance name.

2.785.1.1 IsisUloopModel

cfgDelayV4	*integer*	IPv4 local convergence delay.
cfgDelayV6	*integer*	IPv6 local convergence delay.
delayAbortsL1	*integer*	Number of failures at level-1.
delayAbortsL2	*integer*	Number of failures at level-2.
delayAttemptsL1	*integer*	Number of attempts at level-1.
delayAttemptsL2	*integer*	Number of attempts at level-2.
systemId	*string*	System ID.
uloopSms	*list of IsisUloopSmModel* see 2.785.1.1.1	List of Micro-loop SMs.

2.785.1.1.1 IsisUloopSmModel

adjacencyDownTimestamp **	*integer*	Link or BFD failure timestamp. **Since: 4.25.1
cause	*enum: linkDown | bfdDown*	Reason for Micro-loop prevention attempt.
delayTimeRemainingV4	*integer*	Delay time remaining for IPv4 routes.
delayTimeRemainingV6	*integer*	Delay time remaining for IPv6 routes.
delayTimerStartedV4	*integer*	Time delay timer started for IPv4 routes.
delayTimerStartedV6	*integer*	Time delay timer started for IPv6 routes.
delayTimerStoppedV4	*integer*	Time delay timer stopped for IPv4 routes.
delayTimerStoppedV6	*integer*	Time delay timer stopped for IPv6 routes.
delayV4	*integer*	IPv4 local convergence delay in msecs.
delayV6	*integer*	IPv6 local convergence delay in msecs.
[delayedRoutesV4]	*list of IsisUloopDelayedV4-RouteModel* see 2.785.1.1.1.1	List of IPv4 routes delayed.
[delayedRoutesV6]	*list of IsisUloopDelayedV6-RouteModel* see 2.785.1.1.1.2	List of IPv6 routes delayed.
intfIndex	*integer*	Interface index.
intfName	*interface name*	Protected interface.
lastAttempt	*boolean*	Last attempt.
level	*integer*	Level.
protectionV4	*enum: disabled | node | link*	TI-LFA protection mode for IPv4.
protectionV6	*enum: disabled | node | link*	TI-LFA protection mode for IPv6.

result	*enum: delayAbortedLinkDown | delayAbortedCleanup | delayAbortedDelayRoutes-NewSpf | delaySucceeded | delayAbortedSpfNot-Scheduled | delayAbortedSpfScheduled | delayAbortedLinkUp | delayAbortedDelayRoutes | delayAbortedStateStarted*	Result of Micro-loop prevention attempt.
routesDelayedV4	*integer*	Number of IPv4 routes delayed.
routesDelayedV6	*integer*	Number of IPv6 routes delayed.
state	*enum: delaySpfScheduled | delayNotRunnng | delayStarted | delayRoutesNewSpf | delayRoutes*	State of Micro-loop prevention attempt.

2.785.1.1.1.1 IsisUloopDelayedV4RouteModel

[algorithm] **	*string*	Algorithm identifier. **Since: **currentRelease**
maskLength	*integer*	Mask Length.
v4DestAddr	*IPv4 address*	IPv4 destination address of delayed route.

2.785.1.1.1.2 IsisUloopDelayedV6RouteModel

[algorithm] **	*string*	Algorithm identifier. **Since: **currentRelease**
maskLength	*integer*	Mask Length.
v6DestAddr	*IPv6 address*	IPv6 destination address of delayed route.

2.786 show isis (lsp log) | (INSTANCE lsp log) | (lsp log vrf VRFNAME)

vrfs	*table(string) = Instances-IsisLspLogTableModel* see 2.786.1	IS-IS instance information for all VRFs.

2.786.1 InstancesIsisLspLogTableModel

isisInstances	*table(string) = IsisLspLogTableModel* see 2.786.1.1	IS-IS Instance list keyed by instance name.

2.786.1.1 IsisLspLogTableModel

lspLogs	*list of IsisLspLogEntryModel* see 2.786.1.1.1	List of lsplog Entries.

2.786.1.1.1 IsisLspLogEntryModel

event	*enum: backoff restarted, hold value | backoff continue, remaining wait time | out-delay applied, delayed by | backoff started, hold value*	Event associated with Log.
holdInterval	*integer*	Hold interval for the LSP in ms.
logId	*integer*	Log entry identifier.
logTime	*integer*	Log time stamp.
lspId	*string*	IS-IS LSPID.
lspLevel	*integer*	Level of the LSP.

2.787 show isis (lsp purges) | (INSTANCE lsp purges) | (lsp purges vrf VRFNAME) [...]

vrfs	*table(string) = Instances-IsisLspPurgesModel* see 2.787.1	IS-IS instance information for all VRFs.

2.787.1 InstancesIsisLspPurgesModel

isisInstances	*table(string) = IsisLspPurgesModel* see 2.787.1.1	IS-IS Instance list keyed by instance name.

2.787.1.1 IsisLspPurgesModel

levels	*table(numeric string) = IsisLevelLspPurgesModel* see 2.787.1.1.1	Dictionary of purges keyed by level.

2.787.1.1.1 IsisLevelLspPurgesModel

purgeLogs	*list of IsisLspPurgesEntryModel* see 2.787.1.1.1.1	List of IS-IS purge entries.

2.787.1.1.1.1 IsisLspPurgesEntryModel

logTime	*floating point number*	Log time stamp.
lspId	*string*	IS-IS LSPID of purge.
[lspIdWithHostname]	*string*	LSP generator hostname.
[originatorHostname]	*string*	Purge Originator or receiver Dynamic Hostname.
[originatorSystemId]	*string*	Purge Originator or receiver SystemID.

[reason]	*enum: obsoleteProxyLsp | differentChecksum | disResigned | userClearedLsp | lspLifetimeExpired | receivedOldIncarnation | noContents*	Reason for purge generation.
[upstreamNgbHostname]	*string*	Neighbour Hostname who send purge.
[upstreamNgbSystemId]	*string*	Neighbour SystemID who send purge.

2.788 show isis (network topology) | (INSTANCE network topology) | (network topology vrf VRFNAME) [...]

vrfs	*table(string) = Instances-IsisTopologyModel* see 2.788.1	Topology information for all VRFs.

2.788.1 InstancesIsisTopologyModel

isisInstances	*table(string) = IsisTopologyModel* see 2.788.1.1	IS-IS Instance list keyed by instance name.

2.788.1.1 IsisTopologyModel

levels	*table(numeric string) = IsisTopologyEntryModel* see 2.788.1.1.1	A dict of isis Topology keyed by isis level.

2.788.1.1.1 IsisTopologyEntryModel

topologies	*table(string) = IsisTopologyNextHopModel* see 2.788.1.1.1.1	Dict of address-families keyed by topology-id.

2.788.1.1.1.1 IsisTopologyNextHopModel

nexthops	*table(string) = IsisTopologyNextHops-EntryModel* see 2.788.1.1.1.2	Dict of next-hops keyed by system-id.

2.788.1.1.1.2 IsisTopologyNextHopsEntryModel

[destinationHostname]	*string*	Hostname.
vias	*list of IsisTopologyEntryListModel* see 2.788.1.1.1.3	Isis nexthops list.

2.788.1.1.1.3 IsisTopologyEntryListModel

`iaMetric` **	*integer*	IS-IS inter-area metric value. **Since: 4.24.0
`[igpShortcutTunnel]` **	*TunnelInfo* see 2.151.1.3.1	IGP Shortcut tunnel. **Since: 4.25.1
`interface`	*interface name*	IS-IS interface name.
`metric`	*integer*	IS-IS metric value.
`[nexthopHostname]`	*string*	Hostname.
`nexthopSystemId`	*string*	Next Hop.
`snpa`	*string*	Subnetwork Point of Attachment of Neighbor.

2.789 show isis (segment-routing) | (INSTANCE segment-routing) | (segment-routing vrf VRFNAME)

vrfs	*table(string) = InstancesSrSummaryModel* see 2.789.1	IS-IS instance information for all VRFs.

2.789.1 InstancesSrSummaryModel

isisInstances	*table(string) = SrSummaryModel* see 2.789.1.1	IS-IS Instance list keyed by instance name.

2.789.1.1 SrSummaryModel

adjSidAllocationMode	*enum: SrOnly | All | None*	Adj-SID allocation mode.
[adjSidPoolBase]	*integer*	Base of Adj-SID pool.
[adjSidPoolSize]	*integer*	Size of Adj-SID pool.
dataPlane	*enum: BOTH | IPV6 | None | MPLS*	Data-plane.
[hostname] **	*string*	Hostname. **Since: 4.20.0
inspectProxyAttachedFlag **	*boolean*	Inspect attached-flag for proxy-node segments. **Since: **currentRelease**
mappingServer	*boolean*	Set if this is a Segment Routing Mapping Server (SRMS).
reachabilityAlgorithm	*enum: SSPF | SPF*	IS-IS Reachability Algorithm.
[routerId]	*IPv4 address*	Router ID.
selfOriginatedSegment-Statistics	*SelfOriginatedSegment-StatisticsModel* see 2.789.1.1.1	Self-originated segment statistics.
srPeerCount	*integer*	Number of IS-IS SR capable peers.
srgbBase	*integer*	Base of SRGB.
srgbSize	*integer*	Size of SRGB.
systemId	*string*	System ID.

2.789.1.1.1 SelfOriginatedSegmentStatisticsModel

adjSidCount	*integer*	Number of IS-IS SR adjacency segments.
nodeSidCount	*integer*	Number of SR node segments.
prefixSidCount	*integer*	Number of SR prefix segments.
proxyNodeSidCount	*integer*	Number of SR proxy node segments.

2.790 show isis (segment-routing adjacency-segments) | (INSTANCE segment-routing adjacency-segments) | (segment-routing adjacency-segments vrf VRFNAME)

vrfs	*table(string) = InstancesAdjacency-SegmentSummaryModel* see 2.790.1	IS-IS instance information for all VRFs.

2.790.1 InstancesAdjacencySegmentSummaryModel

isisInstances	*table(string) = Adjacency-SegmentSummaryModel* see 2.790.1.1	IS-IS Instance list keyed by instance name.

2.790.1.1 AdjacencySegmentSummaryModel

adjSidAllocationMode	*enum: SrOnly | All | None*	Adj-SID allocation mode.
[adjSidPoolBase]	*integer*	Base of Adj-SID pool.
[adjSidPoolSize]	*integer*	Size of Adj-SID pool.
adjacencySegments	*list of AdjacencySegmentModel* see 2.790.1.1.1	List of locally originated Adjacency Segments.
dataPlane	*enum: BOTH | IPV6 | None | MPLS*	Data-plane.
[hostname] **	*string*	Hostname. **Since: 4.20.0
misconfiguredAdjacency-Segments **	*list of Misconfigured-AdjacencySegmentModel* see 2.790.1.1.2	List of misconfigured Adjacency Segments. **Since: 4.20.0
receivedGlobalAdjacency-Segments **	*list of ReceivedGlobal-AdjacencySegmentModel* see 2.790.1.1.3	List of received global Adjacency Segments. **Since: 4.21.0
[routerId]	*IPv4 address*	Router ID.
systemId	*string*	System ID.

2.790.1.1.1 AdjacencySegmentModel

flags **	*AdjacencySegmentFlags-Model* see 2.790.1.1.1.1	Flags. **Since: 4.20.5
ipAddress	*IP address*	IP Address of the adjacency.
lan	*boolean*	This is a LAN-SID.
level **	*integer*	Level. **Since: 4.20.5
localIntf	*interface name*	Local interface.
protection **	*enum: node | link | unprotected*	Protection mode of the local adj-SID. **Since: 4.22.1
sid	*integer*	Adjacency Segment assigned to this adjacency.

sidOrigin **	*enum: configured | dynamic*	Indicates if adj-SID is configured manually or assigned dynamically. **Since: 4.20.0
[srlgProtection] **	*enum: loose | strict*	TI-LFA SRLG protection mode of the segment. **Since: 4.24.1

2.790.1.1.1.1 AdjacencySegmentFlagsModel

b	*boolean*	Adj-SID is eligible for protection.
f	*boolean*	Adj-SID refers to an adjacency with outgoing IPv6 encapsulation.
l	*boolean*	Adj-SID has local significance.
s	*boolean*	Adj-SID refers to a set of adjacencies.
v	*boolean*	Adj-SID carries a value.

2.790.1.1.2 MisconfiguredAdjacencySegmentModel

af	*enum: ipv4 | ipv6*	Address Family.
localIntf	*interface name*	Local interface.
reason	*enum: interface-type-mismatch | out-of-range*	Reason why this configured adjacency segments is invalid.
sid	*integer*	Adjacency Segment assigned to this adjacency.

2.790.1.1.3 ReceivedGlobalAdjacencySegmentModel

flags	*AdjacencySegmentFlagsModel* see 2.790.1.1.1.1	Flags.
[hostname]	*string*	Hostname of the segment originator.
[ngbHostname]	*string*	IS-IS neighbor hostname of the segment originator.
ngbSystemId	*string*	IS-IS neighbor of the segment originator.
protection **	*enum: node | link | unprotected*	Protection mode of the global adj-SID. **Since: 4.22.1
sid	*integer*	Segment Id assigned to this adjacency segment.
srReachability **	*enum: srPartiallyReachable | srReachable | srUnreachable*	Reachability information for Adjacency Segments. **Since: 4.25.2
[srlgProtection] **	*enum: loose | strict*	TI-LFA SRLG protection mode of the segment. **Since: 4.24.1
systemId	*string*	System ID of the segment originator.

2.791 show isis (segment-routing global-blocks) | (INSTANCE segment-routing global-blocks) | (segment-routing global-blocks vrf VRF-NAME)

vrfs	*table(string) = Instances-GlobalBlockSummaryModel* see 2.791.1	IS-IS instance information for all VRFs.

2.791.1 InstancesGlobalBlockSummaryModel

isisInstances	*table(string) = GlobalBlockSummaryModel* see 2.791.1.1	IS-IS Instance list keyed by instance name.

2.791.1.1 GlobalBlockSummaryModel

dataPlane	*enum: BOTH | IPV6 | None | MPLS*	Data-plane.
globalBlocks	*list of GlobalBlockModel* see 2.791.1.1.1	Global blocks.
[hostname] **	*string*	Hostname. **Since: 4.20.0
[routerId]	*IPv4 address*	Router ID.
srPeerCount	*integer*	Number of IS-IS SR peers capable segments.
srgbBase	*integer*	Base of SRGB.
srgbSize	*integer*	Size of SRGB.
systemId	*string*	System ID.

2.791.1.1.1 GlobalBlockModel

base	*integer*	SRGB base.
[hostname] **	*string*	Hostname. **Since: 4.20.0
size	*integer*	SRGB size.
systemId	*string*	System ID.

2.792 show isis (segment-routing prefix-segments) | (INSTANCE segment-routing prefix-segments) | (segment-routing prefix-segments vrf VRF-NAME) [...]

revision 2

vrfs	*table(string) = InstancesPrefixSegment-SummaryModel* see 2.792.1	IS-IS instance information for all VRFs.

2.792.1 InstancesPrefixSegmentSummaryModel

isisInstances	*table(string) = Prefix-SegmentSummaryModel* see 2.792.1.1	IS-IS Instance list keyed by instance name.

2.792.1.1 PrefixSegmentSummaryModel

dataPlane	*enum: BOTH | IPV6 | None | MPLS*	Data-plane.
[hostname]	*string*	Hostname.
nodeSidCount	*integer*	Number of SR node segments.
prefixSegments	*list of PrefixSegmentModel* see 2.792.1.1.1	Prefix segments.
prefixSidCount	*integer*	Number of SR prefix segments.
proxyNodeSidCount	*integer*	Number of SR proxy node segments.
[routerId]	*IPv4 address*	Router ID.
systemId	*string*	System ID.

2.792.1.1.1 PrefixSegmentModel

[algorithm] **	*string*	Algorithm of the prefix segment. **Since: 4.26.2
flags	*PrefixSegmentFlagsModel* see 2.792.1.1.1.1	Flags.
[hostname]	*string*	Hostname.
level	*integer*	Level.
prefix	*IP address prefix*	Prefix address with netmask.
protection	*enum: node | link | unprotected*	Protection mode of the prefix segment.
[proxyNodeFlags] **	*ProxyNodeSegmentFlags-Model* see 2.792.1.1.1.2	proxyNodeFlags. **Since: **currentRelease**
segmentId	*integer*	Segment ID.
segmentType	*enum: Proxy-Node | Node | Prefix*	Type.
selfOriginatedPrefix	*boolean*	Prefix is self originated.
srReachability	*enum: srPartiallyReachable | srReachable | srUnreachable*	Reachability information for Prefix Segments.

[srlgProtection]	*enum: loose | strict*	TI-LFA SRLG protection mode of the segment.
[systemId]	*string*	System identifier.
[systemIds]	*list of string*	System IDs sharing anycast segment.

2.792.1.1.1.1 PrefixSegmentFlagsModel

e	*boolean*	Set if any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with a Prefix-SID having an Explicit-NULL value before forwarding the packet.
l	*boolean*	Set if the value/index carried by the Prefix-SID has local significance.
n	*boolean*	Set if the Prefix-SID refers to the router identified by the prefix.
p	*boolean*	Set if the penultimate hop MUST NOT pop the Prefix-SID before delivering the packet to the node that advertised the Prefix-SID.
r	*boolean*	Set if prefix to which this Prefix-SID is attached has been propagated by the router either from another level or from redistribution.
v	*boolean*	Set if the Adj-SID carries a value.

2.792.1.1.1.2 ProxyNodeSegmentFlagsModel

a	*boolean*	Set if the proxy-node segment is directly connected to the node advertising the prefix.

2.793 show isis (spf log) | (INSTANCE spf log) | (spf log vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisSpfLogTableModel* see 2.793.1	IS-IS instance information for all VRFs.

2.793.1 InstancesIsisSpfLogTableModel

isisInstances	*table(string) = IsisSpfLogTableModel* see 2.793.1.1	IS-IS Instance list keyed by instance name.

2.793.1.1 IsisSpfLogTableModel

levels	*table(numeric string) = IsisLevelSpfLogModel* see 2.793.1.1.1	Dictionary of SPF logs keyed by level.

2.793.1.1.1 IsisLevelSpfLogModel

spfLogs	*list of IsisSpfLogEntry* see 2.793.1.1.1.1	ISIS SPF log entries.

2.793.1.1.1.1 IsisSpfLogEntry

[ipv4Nodes]	*integer*	Number of IPv4 routers that make up the topology.
[ipv6Nodes]	*integer*	Number of IPv6 routers that make up the topology.
[partial]	*boolean*	Partial SPF optimization.
reason	*string*	The first trigger that scheduled the SPF.
[scheduled]	*boolean*	Scheduled SPF run.
startTime	*floating point number*	SPF start time in UTC.
[timeTaken]	*floating point number*	SPF run duration(ms).
triggers	*integer*	Number of SPF triggers consumed in this SPF run.

2.794 show isis (summary-address) | (INSTANCE summary-address) | (summary-address vrf VRFNAME) [...]

vrfs	*table(string) = InstancesIsisLeakedRouteSumAddrInstanceModel* see 2.794.1	ISIS route summary-address for all VRFs.

2.794.1 InstancesIsisLeakedRouteSumAddrInstanceModel

isisInstances	*table(string) = IsisLeakedRouteSumAddrInstanceModel* see 2.794.1.1	IS-IS Instance list keyed by instance name.

2.794.1.1 IsisLeakedRouteSumAddrInstanceModel

level1ToLevel2SummaryAddresses	*table(IPv4 or IPv6 address prefix) = IsisSumAddrEntryModel* see 2.794.1.1.1	A dictionary of summary-addresses for L1 to L2 leaked routes.

2.794.1.1.1 IsisSumAddrEntryModel

advertised	*string*	Summary address advertised.
[contributors]	*list of IP address prefix*	List of contributors.
metric	*integer*	Metric value.
totalContributors	*integer*	Total number of contributors.

2.795 show isis counters | (INSTANCE counters) | (counters vrf VRF-NAME) [...]

vrfs	*table(string) = Instances-IsisInstanceCounters* see 2.795.1	ISIS instance information for all VRFs.

2.795.1 InstancesIsisInstanceCounters

isisInstances	*table(string) = IsisInstanceCounters* see 2.795.1.1	IS-IS Instance list keyed by instance name.

2.795.1.1 IsisInstanceCounters

interfaceCounters	*table(Interface name) = IsisInterfaceCounters* see 2.795.1.1.1	Dictionary of Isis Interface counterskeyed by interface name.
[systemCounters] **	*IsisSystemCounters* see 2.795.1.1.2	Data pertaining to IS-IS System counters. **Since: 4.28.0

2.795.1.1.1 IsisInterfaceCounters

[dropCounters]	*IsisInterfaceDropCounters* see 2.795.1.1.1.1	Data pertaining to IS-IS drop counters.
[lastResetTime] **	*integer*	UTC timestamp when the counters are reset last. **Since: 4.21.3
[packetCounters]	*IsisInterfacePacketCounters* see 2.795.1.1.1.13	Data pertaining to the packet counters.

2.795.1.1.1.1 IsisInterfaceDropCounters

csnpL1	*integer*	Number of Level-1 IS-IS Complete Sequence Number Packets dropped.
csnpL2	*integer*	Number of Level-2 IS-IS Complete Sequence Number Packets dropped.
[details]	*IsisInterfaceDropCounters-Details* see 2.795.1.1.1.2	Reasons ofdrop for each PDU with their respective counts.
iihLanL1	*integer*	Number of Level-1 IS-IS Lan Hello Packets Dropped.
iihLanL2	*integer*	Number of Level-2 IS-IS Lan Hello Packets Dropped.
iihP2p	*integer*	Number of IS-IS P2P Hello Packets Dropped.
lspL1	*integer*	Number of Level-1 IS-IS Link State PDUs dropped.
lspL2	*integer*	Number of Level-2 IS-IS Link State PDUs dropped.

malformedPdus	*integer*	Number of IS-IS PDUs received malformed. The received PDU had either a wrong header length, wrong version, or the System Id length in the PDU is not supported by the system received.
psnpL1	*integer*	Number of Level-1 IS-IS Partial Sequence Number Packets dropped.
psnpL2	*integer*	Number of Level-2 IS-IS Partial Sequence Number Packets dropped.

2.795.1.1.1.2 IsisInterfaceDropCountersDetails

[csnpL1DropReasons]	*IsisInterfaceDropReasons-L1Csnps* see 2.795.1.1.1.3	Reasons which led to Level-1 IS-IS Complete Sequence number packets getting dropped and their respective counts.
[csnpL2DropReasons]	*IsisInterfaceDropReasons-L2Csnps* see 2.795.1.1.1.7	Reasons which led to Level-2 IS-IS Complete Sequence number packets getting dropped and their respective counts.
[iihLanL1DropReasons]	*IsisInterfaceDropReasons-L1LanHellos* see 2.795.1.1.1.4	Reasons which led to Level-1 IS-IS LAN Hellosgetting dropped and their respective counts.
[iihLanL2DropReasons]	*IsisInterfaceDropReasons-L2LanHellos* see 2.795.1.1.1.8	Reasons which led to Level-2 IS-IS LAN Hellosgetting dropped and their respective counts.
[iihP2pDropReasons]	*IsisInterfaceDropReasons-P2pHellos* see 2.795.1.1.1.12	Reasons which led IS-IS P2P Hellosgetting dropped and their respective counts.
[lspL1DropReasons]	*IsisInterfaceDropReasons-L1Lsps* see 2.795.1.1.1.5	Reasons which led to IS-IS Level-1 Link State PDUs getting dropped and their respective counts.
[lspL2DropReasons]	*IsisInterfaceDropReasons-L2Lsps* see 2.795.1.1.1.9	Reasons which led to IS-IS Level-2 Link State PDUs getting dropped and their respective counts.
[malformedPdusDropReasons]	*IsisInterfaceDropReasons-MalformedPdus* see 2.795.1.1.1.11	Reasons which led to malformed PDUsgetting dropped and their respective counts.
[psnpL1DropReasons]	*IsisInterfaceDropReasons-L1Psnps* see 2.795.1.1.1.6	Reasons which led to IS-IS Level-1 Partial Sequence number packets getting dropped and their respective counts.
[psnpL2DropReasons]	*IsisInterfaceDropReasons-L2Psnps* see 2.795.1.1.1.10	Reasons which led to IS-IS Level-2 Partial Sequence number packets getting dropped and their respective counts.

2.795.1.1.1.3 IsisInterfaceDropReasonsL1Csnps

[authFailure]	*integer*	The PDU could not be authenticated.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.

[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidLspEntryOption]	*integer*	CSNPs and PSNPs are dropped with this reason. one or more of the LSP entry sub-TLVs in the PDU has an invalid length.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[nonAdjSource]	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU.

2.795.1.1.1.4 IsisInterfaceDropReasonsL1LanHellos

[authFailure]	*integer*	The PDU could not be authenticated.
[badAreaOption]	*integer*	Hellos are dropped with this reason. The Area address sub-option in the Areas option is malformed. The areas indicated in the PDU can hence not be used.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[circuitTypeMismatches]	*integer*	The Circuit type indicated in the PDU does not match with that of the interface on which it was received For instance, a p2p Hello received on a IS-IS Broadcast interface is dropped with this reason.
[dupSysIdDetected] **	*integer*	We already have an adjacency with the same system Id. **Since: 4.21.3
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[incompatibleAdjAttrs]	*integer*	Hellos are dropped with this reason. we drop with this reason when there are no usable addresses configured on the interface receiving this Hello OR the protocols supported by our syatem have changed and have become inconsistent with those supported by the system sending this Hello.
[lengthGreaterThanMTU]	*integer*	Length of the PDU is greater than the MTU configured on the interface on which it is received.

[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[maxAreaAddressMismatch]	*integer*	Hellos are dropped with this reason. The max number of areas indicated in the Hello mismatch with the maximum number of areas supported by the system receiving this hello.
[missingAreas]	*integer*	Hellos are dropped with this reason. The IS-IS interface receiving this Hello is on Level-1 but there is no area TLV in the Hello. We hence drop the Hello.
[neighbourSysIdChanged]	*integer*	LAN Hellos are Dropped with this reason. The system Id of the neighbour sending this Hellohas changed. We hence drop the Hello and delete the adjacency.
[ngbSysIdMismatchInRestart-TLVWithRA]	*integer*	Hellos are dropped with this reason. The neighbour system Id indicated in restart TLV with RA does not match our own System Id.
[noMatchingAreas]	*integer*	Hellos are dropped with this reason. The IS-IS interface on which the Hello was received is in level-1 but none of the areas indicated in the Hello match with the areas supported by the system.
[noRemTimeInRestartTlvWith-RA]	*integer*	Hellos are dropped with this reason. Remaining time is not indicated in restart TLV with RA.
[noSourceMac]	*integer*	LAN Hellos are dropped with this reason. The Hello is received without any source MAC address.
[noSupportedProtocols]	*integer*	Hellos are dropped with this reason. The Hello received does not have NLPID TLV indicating that no protocols are supported by the system sending the Hello.
[ownSystemIdPdu]	*integer*	The System Id on the PDU is similar to that of the system on which it is received indicating the system received the PDU it has sent (or) some other system is using the same system ID. The PDU is dropped.
[protocolAddressesMismatch]	*integer*	Hellos are dropped with this reason.The Interface addresses received in the Hello are inconsistent with the NLPIDs received indicating that there is an inconsistency between the protocols supported and interface addresses configured on the system sending this Hello.
[supportedProtocols-Mismatch]	*integer*	Hellos are dropped with this reason.The NLPIDs received in the PDU do not match the protocols supported by the system receiving this PDU.

[topoSuppProtocolsMismatch]	*integer*	Hellos are dropped with this reaosn. This happens when the MTIDs and NLPIDs in the PDU mismatch, indicating that there is an inconsistency between the topologies and protocols supported by the system sending this Hello.
[unknownCircuitType]	*integer*	The circuit type indicated in the PDU is unknown. This Happens when a IS-IS Hello has a circuit type other than 1,2 or 3 indicating Level-1,Level-2,Level-1-2 respectively.

2.795.1.1.1.5 IsisInterfaceDropReasonsL1Lsps

[authFailure]	*integer*	The PDU could not be authenticated.
[badChecksum]	*integer*	The checksum indicated in the LSP is wrong indicating that the Lsp is corrupted.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badLspLength]	*integer*	Length of the LSP is greater than the length of LSP buffer of the receiving system.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[differentChecksumLsp]	*integer*	LSP received has samilar sequence number as that in the LSP database of the system receiving the LSP, but their checksums are different. The LSP is dropped. If the LSP is self generated then it is regenerated with a newer sequence number. If it is not self generated then it is force expired.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidPurge]	*integer*	LSPs are dropped with this reason. The Purge LSP must contain only the Header (or) the Header and Auth option. If it contains anything more or less, it is dropped.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[nonAdjSource]	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU.
[olderLsp]	*integer*	LSP with older sequence number than that of the corresponding LSP in the database of the system is received. The LSP is dropped and the latest LSP is flooded into the circuit on which the LSP was received.

olderPurgeLsp	*integer*	Purge LSP with a sequence number older than that in the LSP database of the system is received. The LSP is dropped and the latest LSP is flooded.
[ownExpiredLsp]	*integer*	LSP is received with the system Id in the LSPID matching the system Id of the system receiving it, but the LSP is no longer present in the LSP database of the system.This LSP might be from the previous incarnation of the system. The LSP is dropped and is force expired.
[ownLspPurge]	*integer*	Self-originated LSP of the system is received with a zero-life but the remaining life of the LSP as indicated in the LSP database is non-zero. The system drops this LSP and regenerates the LSP with a newer sequence number.
[ownNewerSequenceLsp]	*integer*	Self generated LSP with a newer sequence number has been received. It is dropped and a new LSP is regenerated and flooded.
[parsingFailure]	*integer*	LSPs are dropped with this reasons. The options in the LSP could not be parsed properly, indicating that the received LSP is corrupted.

2.795.1.1.1.6 IsisInterfaceDropReasonsL1Psnps

[authFailure]	*integer*	The PDU could not be authenticated.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidLspEntryOption]	*integer*	CSNPs and PSNPs are dropped with this reason. one or more of the LSP entry sub-TLVs in the PDU has an invalid length.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[nonAdjSource] **	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU. **Since: 4.23.0
[psnpOnNonDisSystem]	*integer*	PSNP is received on a circuit of the system of type broadcast but system is not a DIS on that circuit.

2.795.1.1.1.7 IsisInterfaceDropReasonsL2Csnps

[authFailure]	*integer*	The PDU could not be authenticated.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidLspEntryOption]	*integer*	CSNPs and PSNPs are dropped with this reason. one or more of the LSP entry sub-TLVs in the PDU has an invalid length.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[nonAdjSource]	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU.

2.795.1.1.1.8 IsisInterfaceDropReasonsL2LanHellos

[authFailure]	*integer*	The PDU could not be authenticated.
[badAreaOption]	*integer*	Hellos are dropped with this reason. The Area address sub-option in the Areas option is malformed. The areas indicated in the PDU can hence not be used.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[circuitTypeMismatches]	*integer*	The Circuit type indicated in the PDU does not match with that of the interface on which it was received For instance, a p2p Hello received on a IS-IS Broadcast interface is dropped with this reason.
[dupSysIdDetected] **	*integer*	We already have an adjacency with the same system Id. **Since: 4.21.3
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0

[incompatibleAdjAttrs]	*integer*	Hellos are dropped with this reason. we drop with this reason when there are no usable addresses configured on the interface receiving this Hello OR the protocols supported by our syatem have changed and have become inconsistent with those supported by the system sending this Hello.
[lengthGreaterThanMTU]	*integer*	Length of the PDU is greater than the MTU configured on the interface on which it is received.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[maxAreaAddressMismatch]	*integer*	Hellos are dropped with this reason. The max number of areas indicated in the Hello mismatch with the maximum number of areas supported by the system receiving this hello.
[missingAreas]	*integer*	Hellos are dropped with this reason. The IS-IS interface receiving this Hello is on Level-1 but there is no area TLV in the Hello. We hence drop the Hello.
[neighbourSysIdChanged]	*integer*	LAN Hellos are Dropped with this reason. The system Id of the neighbour sending this Hellohas changed. We hence drop the Hello and delete the adjacency.
[ngbSysIdMismatchInRestart-TLVWithRA]	*integer*	Hellos are dropped with this reason. The neighbour system Id indicated in restart TLV with RA does not match our own System Id.
[noMatchingAreas]	*integer*	Hellos are dropped with this reason. The IS-IS interface on which the Hello was received is in level-1 but none of the areas indicated in the Hello match with the areas supported by the system.
[noRemTimeInRestartTlvWith-RA]	*integer*	Hellos are dropped with this reason. Remaining time is not indicated in restart TLV with RA.
[noSourceMac]	*integer*	LAN Hellos are dropped with this reason. The Hello is received without any source MAC address.
[noSupportedProtocols]	*integer*	Hellos are dropped with this reason. The Hello received does not have NLPID TLV indicating that no protocols are supported by the system sending the Hello.
[ownSystemIdPdu]	*integer*	The System Id on the PDU is similar to that of the system on which it is received indicating the system received the PDU it has sent (or) some other system is using the same system ID. The PDU is dropped.

[protocolAddressesMismatch]	*integer*	Hellos are dropped with this reason.The Interface addresses received in the Hello are inconsistent with the NLPIDs received indicating that there is an inconsistency between the protocols supported and interface addresses configured on the system sending this Hello.
[supportedProtocols-Mismatch]	*integer*	Hellos are dropped with this reason.The NLPIDs received in the PDU do not match the protocols supported by the system receiving this PDU.
[topoSuppProtocolsMismatch]	*integer*	Hellos are dropped with this reaosn. This happens when the MTIDs and NLPIDs in the PDU mismatch, indicating that there is an inconsistency between the topologies and protocols supported by the system sending this Hello.
[unknownCircuitType]	*integer*	The circuit type indicated in the PDU is unknown. This Happens when a IS-IS Hello has a circuit type other than 1,2 or 3 indicating Level-1,Level-2,Level-1-2 respectively.

2.795.1.1.1.9 IsisInterfaceDropReasonsL2Lsps

[authFailure]	*integer*	The PDU could not be authenticated.
[badChecksum]	*integer*	The checksum indicated in the LSP is wrong indicating that the Lsp is corrupted.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badLspLength]	*integer*	Length of the LSP is greater than the length of LSP buffer of the receiving system.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[differentChecksumLsp]	*integer*	LSP received has samilar sequence number as that in the LSP database of the system receiving the LSP, but their checksums are different. The LSP is dropped. If the LSP is self generated then it is regenerated with a newer sequence number. If it is not self generated then it is force expired.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidPurge]	*integer*	LSPs are dropped with this reason. The Purge LSP must contain only the Header (or) the Header and Auth option. If it contains anything more or less, it is dropped.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.

[nonAdjSource]	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU.
[olderLsp]	*integer*	LSP with older sequence number than that of the corresponding LSP in the database of the system is received. The LSP is dropped and the latest LSP is flooded into the circuit on which the LSP was received.
olderPurgeLsp	*integer*	Purge LSP with a sequence number older than that in the LSP database of the system is received. The LSP is dropped and the latest LSP is flooded.
[ownExpiredLsp]	*integer*	LSP is received with the system Id in the LSPID matching the system Id of the system receiving it, but the LSP is no longer present in the LSP database of the system.This LSP might be from the previous incarnation of the system. The LSP is dropped and is force expired.
[ownLspPurge]	*integer*	Self-originated LSP of the system is received with a zero-life but the remaining life of the LSP as indicated in the LSP database is non-zero. The system drops this LSP and regenerates the LSP with a newer sequence number.
[ownNewerSequenceLsp]	*integer*	Self generated LSP with a newer sequence number has been received. It is dropped and a new LSP is regenerated and flooded.
[parsingFailure]	*integer*	LSPs are dropped with this reasons. The options in the LSP could not be parsed properly, indicating that the received LSP is corrupted.

2.795.1.1.1.10 IsisInterfaceDropReasonsL2Psnps

[authFailure]	*integer*	The PDU could not be authenticated.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch]	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0
[invalidLspEntryOption]	*integer*	CSNPs and PSNPs are dropped with this reason. one or more of the LSP entry sub-TLVs in the PDU has an invalid length.

[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[nonAdjSource] **	*integer*	LSPs, CSNPs or PSNPs are dropped with this reason. The system receiving this PDU does not have an adjacency with the system sending this PDU. **Since: 4.23.0
[psnpOnNonDisSystem]	*integer*	PSNP is received on a circuit of the system of type broadcast but system is not a DIS on that circuit.

2.795.1.1.1.11 IsisInterfaceDropReasonsMalformedPdus

[idFieldLengthMismatches]	*integer*	System Id Length in the PDU is different from the System Id length supported by the sytem on which the PDU was received.
[shortHeaders]	*integer*	Header length of the IS-IS PDU received is short.
[wrongVersion]	*integer*	Version of the PDU header is incompatible with the version we support.

2.795.1.1.1.12 IsisInterfaceDropReasonsP2pHellos

[authFailure]	*integer*	The PDU could not be authenticated.
[badAdjStateOption]	*integer*	P2P Hellos are dropped with this reason. The option length or the Adjacency state indicated in the option is incompatible with the state of adjacency in which the system receiving this hello currently in.
[badAreaOption]	*integer*	Hellos are dropped with this reason. The Area address sub-option in the Areas option is malformed. The areas indicated in the PDU can hence not be used.
[badLength]	*integer*	Length indicated in the packet header is different from the actual length of the packet.
[badOptionLength]	*integer*	Length of one or more of the optional TLVs in the PDU is different from that indicated in the Length section of that TLV.
[circuitLevelMismatch] **	*integer*	The circuit level indicated in the PDU does not match with the level on which the IS-IS interface receiving this PDU is configured on. **Since: 4.22.1
[circuitTypeMismatches]	*integer*	The Circuit type indicated in the PDU does not match with that of the interface on which it was received For instance, a p2p Hello received on a IS-IS Broadcast interface is dropped with this reason.
[iidTlvMismatch] **	*integer*	Invalid IID-TLV received in the PDU. **Since: 4.30.0

[incompatibleAdjAttrs]	*integer*	Hellos are dropped with this reason. we drop with this reason when there are no usable addresses configured on the interface receiving this Hello OR the protocols supported by our syatem have changed and have become inconsistent with those supported by the system sending this Hello.
[lengthGreaterThanMTU]	*integer*	Length of the PDU is greater than the MTU configured on the interface on which it is received.
[malformedTlv]	*integer*	one of the optional TLVs in the PDU could not be validated.
[maxAreaAddressMismatch]	*integer*	Hellos are dropped with this reason. The max number of areas indicated in the Hello mismatch with the maximum number of areas supported by the system receiving this hello.
[noCommonTopology]	*integer*	Hellos are dropped with this reason. This happens when the system receiving the Hello is in Multi-Topology mode but no MTIDs in the Hello and system are common.
[noMatchingAreas]	*integer*	Hellos are dropped with this reason. The IS-IS interface on which the Hello was received is in level-1 but none of the areas indicated in the Hello match with the areas supported by the system.
[noRemTimeInRestartTlvWith-RA]	*integer*	Hellos are dropped with this reason. Remaining time is not indicated in restart TLV with RA.
[noSupportedProtocols]	*integer*	Hellos are dropped with this reason. The Hello received does not have NLPID TLV indicating that no protocols are supported by the system sending the Hello.
[nonMatchingThreeWay-Handshake]	*integer*	P2P Hellos are dropped with this reason. The neighbour system Id and the Neighbour extended circuit Id indicated in the adjacency state option do not match the system Id and the circuit Id of the system receiving the Hello.
[ownSystemIdPdu]	*integer*	The System Id on the PDU is similar to that of the system on which it is received indicating the system received the PDU it has sent (or) some other system is using the same system ID. The PDU is dropped.
[protocolAddressesMismatch]	*integer*	Hellos are dropped with this reason.The Interface addresses received in the Hello are inconsistent with the NLPIDs received indicating that there is an inconsistency between the protocols supported and interface addresses configured on the system sending this Hello.

[supportedProtocols-Mismatch]	*integer*	Hellos are dropped with this reason.The NLPIDs received in the PDU do not match the protocols supported by the system receiving this PDU.
[topoSuppProtocolsMismatch]	*integer*	Hellos are dropped with this reaosn. This happens when the MTIDs and NLPIDs in the PDU mismatch, indicating that there is an inconsistency between the topologies and protocols supported by the system sending this Hello.
[unknownCircuitType]	*integer*	The circuit type indicated in the PDU is unknown. This Happens when a IS-IS Hello has a circuit type other than 1,2 or 3 indicating Level-1,Level-2,Level-1-2 respectively.

2.795.1.1.1.13 IsisInterfacePacketCounters

processed **	*IsisInterfacePacket-CountersData* see 2.795.1.1.1.14	Processed packet counters. **Since: 4.27.2
received	*IsisInterfacePacket-CountersData* see 2.795.1.1.1.14	Received packet counters.
retransmitted **	*IsisInterfacePacket-CountersRetransmittedData* see 2.795.1.1.1.15	Retransmitted packet counters. **Since: 4.27.2
transmitted	*IsisInterfacePacket-CountersData* see 2.795.1.1.1.14	Transmitted packet counters.

2.795.1.1.1.14 IsisInterfacePacketCountersData

csnpL1	*integer*	Number of L1 CSN Packets.
csnpL2	*integer*	Number of L2 CSN Packets.
iihL1	*integer*	Number of L1 LAN IS-IS Hello Packets.
iihL2	*integer*	Nunber of L2 LAN IS-IS Hello Packets.
iihP2p	*integer*	Number of p2p Hello Packets.
lspL1	*integer*	Number of L1 LSPs.
lspL2	*integer*	Number of L2 LSPs.
psnpL1	*integer*	Number of L1 PSN Packets.
psnpL2	*integer*	Number of L2 PSN Packets.

2.795.1.1.1.15 IsisInterfacePacketCountersRetransmittedData

lspL1	*integer*	Number of L1 LSPs.
lspL2	*integer*	Number of L2 LSPs.

2.795.1.1.2 IsisSystemCounters

systemCountersL1	*IsisSystemCountersData* see 2.795.1.1.2.1	System Counters for Level-1.
systemCountersL2	*IsisSystemCountersData* see 2.795.1.1.2.1	System Counters for Level-2.

2.795.1.1.2.1 IsisSystemCountersData

authKeyFails	*integer*	Number of authentication key failures.
authTypeFails	*integer*	Number of authentication type mismatches.
corruptedLsps	*integer*	Number of corrupted in-memory LSPs detected.
databaseOverloads	*integer*	Number of times the database has become overloaded.
exceededMaxSeqNums	*integer*	Number of times the system has attempted to exceed the maximum sequence number.
idLenMismatches	*integer*	Number of times a PDU is received with a different value for ID field length from that of the receiving system.
lspErrors	*integer*	Number of received LSPs with errors.
maxAreaAddressMismatches	*integer*	Number of times a PDU is received with a different value for MaximumAreaAddresses from that of the receiving system.
ownLspPurges	*integer*	Number of times a zero-aged copy of the system's own LSP is received from some other node.
seqNumSkips	*integer*	Number of times a seq number skip has occurred.
spfRuns	*integer*	Number of times SPF was run for this level.

2.796 show isis database | (INSTANCE database) | (database vrf VRF-NAME) [...]

vrfs	*table(string) = InstancesIsisLsdbSummaryModel* see 2.796.1	IS-IS instance information for all VRFs.

2.796.1 InstancesIsisLsdbSummaryModel

isisInstances	*table(string) = IsisLsdbSummaryModel* see 2.796.1.1	IS-IS Instance list keyed by instance name.

2.796.1.1 IsisLsdbSummaryModel

level	*table(numeric string) = LevelSummaryModel* see 2.796.1.1.1	A mapping of isis level to LSDB.

2.796.1.1.1 LevelSummaryModel

`lsps`	*table(string) = LspBaseModel* see 2.796.1.1.1.1	A mapping of systemId to LSP in LSDB.

2.796.1.1.1.1 LspBaseModel

Could also be an instance of:
LspDetailModel 2.778.1.1.1.1

`checksum`	*integer*	Checksum value.
`expiryTime`	*floating point number*	UTC time for Future Expiry time of LSP.
`[flags]`	*LspFlags* see 2.778.1.1.1.12	IS-IS Lsp flags.
`[intermediateSystemType]`	*enum: L2 | none | L1*	IS-type for LSP.
`length` **	*integer*	LSP length. **Since: 4.27.1
`sequence`	*integer*	Sequence number for this LSP.

2.797 show isis flex-algo path [...]

vrfs	*table(string) = FlexAlgoPathVrfModel* see 2.797.1	A mapping between VRF and information of all flex algo paths in that VRF.

2.797.1 FlexAlgoPathVrfModel

[v4Info]	*FlexAlgoPathAfModel* see 2.797.1.1	Flex algo path information for IPv4 address family.
[v6Info]	*FlexAlgoPathAfModel* see 2.797.1.1	Flex algo path information for IPv6 address family.

2.797.1.1 FlexAlgoPathAfModel

topologies	*table(numeric string) = FlexAlgoPathTopoIdModel* see 2.797.1.1.1	A mapping of topology ID to all flex algo paths.

2.797.1.1.1 FlexAlgoPathTopoIdModel

destinations	*table(string) = FlexAlgoPathDestModel* see 2.797.1.1.1.1	A mapping of path destination to all flex algo paths for that destination.

2.797.1.1.1.1 FlexAlgoPathDestModel

[hostname]	*string*	Hostname.
paths	*table(numeric string) = FlexAlgoPathEntryModel* see 2.797.1.1.1.2	A mapping of an algorithm to the corresponding flex algo path.

2.797.1.1.1.2 FlexAlgoPathEntryModel

[algoName]	*string*	Algorithm name.
[details]	*FlexAlgoPathEntryDetailModel* see 2.797.1.1.1.3	Detailed path information.
vias	*list of FlexAlgoViaModel* see 2.797.1.1.1.5	List of flex algo vias.

2.797.1.1.1.3 FlexAlgoPathEntryDetailModel

changeCount	*integer*	Number of times path updated.
constraint	*FlexAlgoPathConstraintModel* see 2.797.1.1.1.4	Path constraint.
lastUpdatedTime	*floating point number*	UTC timestamp of the last path update.
[metric]	*integer*	Path metric.
pathId	*integer*	Path ID number.
refreshReqSeq	*integer*	Request sequence number.

refreshRespSeq	*integer*	Response sequence number.

2.797.1.1.1.4 FlexAlgoPathConstraintModel

[excludeAdminGroup]	*integer*	Admin group excluded.
[excludeSrlg]	*list of integer*	List of SRLG IDs excluded.
[includeAllAdminGroup]	*integer*	Admin group included.
[includeAnyAdminGroup]	*integer*	Any admin group included.
metricType	*enum: igp | minDelay | te*	Metric type.

2.797.1.1.1.5 FlexAlgoViaModel

intf	*interface name*	L3 interface of next hop.
nexthop	*IP address*	Next hop IP address.

2.798 show isis hostname | (INSTANCE hostname) | (hostname vrf VRF-NAME)

vrfs	*table(string) = InstancesIsisHostname-InstanceModel* see 2.798.1	Hostname information for all VRFs.

2.798.1 InstancesIsisHostnameInstanceModel

isisInstances	*table(string) = Isis-HostnameInstanceModel* see 2.798.1.1	IS-IS Instance list keyed by instance name.

2.798.1.1 IsisHostnameInstanceModel

systemIds	*table(string) = IsisHostnameEntryModel* see 2.798.1.1.1	Dictionary of IS-IS hostnames keyed by SystemID.

2.798.1.1.1 IsisHostnameEntryModel

hostname	*string*	Hostname.
level	*enum: L2 | L1L2 | L1*	IS type.

2.799 show isis interface | (INSTANCE interface) | (interface vrf VRF-NAME) | (interface intf) [...]

vrfs	*table(string) = InstancesIsisInterfaceModel* see 2.799.1	IS-IS interface information for all VRFs.

2.799.1 InstancesIsisInterfaceModel

isisInstances	*table(string) = IsisInterfaceModel* see 2.799.1.1	IS-IS Instance list keyed by instance name.

2.799.1.1 IsisInterfaceModel

interfaces	*table(Interface name) = IsisInterface* see 2.799.1.1.1	Dictionary of IS-IS interfaces keyed by interface name.

2.799.1.1.1 IsisInterface

[areaProxyBoundary] **	*boolean*	Interface enabled for area proxy boundary. **Since: 4.23.2
[bfdIpv4Enabled]	*boolean*	Whether IPv4 bfd is enabled on the interface.
[bfdIpv6Enabled]	*boolean*	Whether IPv6 bfd is enabled on the interface.
[disabledReason]	*DisabledReason* see 2.799.1.1.1.1	Reason for interface being disabled.
enabled	*boolean*	Interface enabled or not.
[helloPaddingEnabled]	*boolean*	Whether hello padding is enabled on the interface.
[index]	*integer*	Interface index.
[interfaceAddressFamily] **	*enum: ipv4 | none | both | ipv6*	Address Family configured on this interface none : Address Family not configured on interface ipv4 : IPv4 configured on interface ipv6 : IPv6 configured on interface both : IPv4 and IPv6 configured on interface. **Since: 4.26.0
[interfaceSpeed]	*integer*	Interface speed in mbps.
[interfaceType]	*enum: loopback | broadcast | point-to-point*	Interface type.
[intfLevels]	*table(string) = IsisInterfaceLevelModel* see 2.799.1.1.1.2	Dictionary of levels of the interface keyed by level name.
[ipv4MetricProfile]	*string*	IPv4 metric profile.
[ipv4RouteTag] **	*integer*	IPv4 route tag. **Since: 4.21.3
[ipv6MetricProfile]	*string*	IPv6 metric profile.
[ipv6RouteTag] **	*integer*	IPv6 route tag. **Since: 4.21.3
[mtu]	*integer*	MTU.

[snpa]	*string*	Subnetwork point of attachment.
[srIndexV4]	*integer*	Node segment index for IPv4.
[srIndexV6]	*integer*	Node segment index for IPv6.
[srNodeSegments] **	*list of IsisNodeSegmentModel* see 2.799.1.1.1.7	Node segments for flex-algo. **Since: 4.26.0
[topology]	*enum: ipv4 | ipv6 | ipv4, ipv6*	Type of topologies enabled.

2.799.1.1.1.1 DisabledReason

circuitType	*string*	Configured circuit type on interface.
message	*string*	Reason for interface being disabled.
routerCircType	*string*	Circuit type of router instance.

2.799.1.1.1.2 IsisInterfaceLevelModel

[authRxDisabled] **	*boolean*	Received hello authentication check is disabled. **Since: 4.22.1
[authenticationMode]	*enum: MD5 | SHA | Text*	Authentication mode.
[authenticationModeKeyId] **	*integer*	Authentication mode key ID. **Since: 4.21.3
[broadcastCircuitInfo]	*BroadcastCircuitInfoModel* see 2.799.1.1.1.3	Broadcast circuit information.
[grStatus]	*GracefulRestartStatus* see 2.799.1.1.1.4	Status of graceful restart.
ipv4Metric	*integer*	IPv4 metric.
[ipv6Metric]	*integer*	IPv6 metric.
isisAdjacencies	*list of IsisLevelAdjModel* see 2.799.1.1.1.5	List of ISIS adjacency entries.
[linkId]	*string*	Link ID for P2P interface.
[numAdjacencies]	*integer*	Number of adjacencies (except for loopback interfaces).
passive **	*boolean*	Passive state. **Since: 4.21.3
[sharedSecretProfile] **	*string*	Shared secret profile. **Since: 4.30.0
v4Protection **	*enum: disabled | node | link*	TI-LFA protection mode for IPv4. **Since: 4.22.1
[v4SrlgProtection] **	*enum: loose | strict*	TI-LFA SRLG protection mode for IPv4. **Since: 4.24.1
v6Protection **	*enum: disabled | node | link*	TI-LFA protection mode for IPv6. **Since: 4.22.1
[v6SrlgProtection] **	*enum: loose | strict*	TI-LFA SRLG protection mode for IPv6. **Since: 4.24.1

2.799.1.1.1.3 BroadcastCircuitInfoModel

dis	*string*	System ID of DIS for broadcast circuit.

[disHostname]	*string*	Hostname of DIS for broadcast circuit.
disPriority	*integer*	DIS Priority for broadcast circuit.
lanId	*string*	LAN ID for broadcast circuit.
[lanIdWithHostname]	*string*	Lan ID with Hostname for broadcast circuit.
priority	*integer*	Priority for broadcast circuit.

2.799.1.1.1.4 GracefulRestartStatus

csnpRcvd	*boolean*	CSNP received.
raRcvd	*boolean*	Restart Acknowledgement received.
rrSent	*boolean*	Restart Request Sent.
saSent	*boolean*	Suppress Adjacency Sent.

2.799.1.1.1.5 IsisLevelAdjModel

addrFamily	*enum: unknown* | *ipv4* | *ipv6* | *both* | *none* | *ipv4, ipv6*	Address Family advertised by neighbor none : Address Family not advertised by neighbor ipv4 : IPv4 advertised by neighbor ipv6 : IPv6 advertised by neighbor both : IPv4 and IPv6 advertised by neighbor.
adjAddrFamilyMatch **	*boolean*	Address families should match for adjacency formation. **Since: 4.26.0
adjType	*enum: unknown* | *l1* | *es* | *l1l2* | *l2*	Adjacency type.
holdTime	*integer*	Hold time in seconds.
[hostname]	*string*	Hostname.
intfAreas	*list of IsisIntfAreaModel* see 2.799.1.1.1.6	List of interface areas.
[ipv4IntfAddr]	*IPv4 address*	IPv4 interface address.
[ipv6IntfAddr]	*IPv6 address*	IPv6 interface address.
level	*enum: unknown* | *level2* | *level1-2* | *level1*	Level.
[priority]	*integer*	Priority.
[snpa]	*string*	Subnetwork point of attachment of neighbor.
state	*enum: down* | *up* | *init* | *failed*	Adjacency state.
[supportedTopologies]	*enum: unknown* | *ipv4* | *ipv6* | *both* | *none* | *ipv4, ipv6*	Supported topologies.
systemId	*string*	System ID of neighbor.

2.799.1.1.1.6 IsisIntfAreaModel

areaId	*string*	Area ID.

2.799.1.1.1.7 IsisNodeSegmentModel

[flexAlgo]	*string*	Flexible algorithm name.
index	*integer*	Node segment index.
ipv6	*boolean*	Whether node segment index is for IPv6.

2.800 show isis mpls ldp sync [...]

vrfs	*table(string) = Instances-IsisMplsLdpSyncModel* see 2.800.1	MPLS LDP sync information for all vrfs.

2.800.1 InstancesIsisMplsLdpSyncModel

isisInstances	*table(string) = IsisMplsLdpSyncModel* see 2.800.1.1	IS-IS Instance list keyed by instance name.

2.800.1.1 IsisMplsLdpSyncModel

interfaces	*table(Interface name) = IsisMplsLdpSyncInterface-Model* see 2.800.1.1.1	MPLS LDP sync information keyed by interface name.

2.800.1.1.1 IsisMplsLdpSyncInterfaceModel

[holddownTime]	*integer*	Holddown time in seconds. Hold until established if zero.
[igpNotifyDelay]	*integer*	IGP notify delay in seconds. Immediate if zero.
ldpEnabled	*boolean*	LDP Enabled.
[syncAchieved]	*boolean*	LDP sync achieved.
syncRequired	*boolean*	LDP sync required.

2.801 show isis neighbors | (INSTANCE neighbors) | (neighbors vrf VRF-NAME) [...]

vrfs	*table(string) = InstancesIsisInstanceNeighbors* see 2.801.1	ISIS instance information for all VRFs.

2.801.1 InstancesIsisInstanceNeighbors

isisInstances	*table(string) = IsisInstanceNeighbors* see 2.801.1.1	IS-IS Instance list keyed by instance name.

2.801.1.1 IsisInstanceNeighbors

neighbors	*table(string) = IsisSystemIdNeighbors* see 2.801.1.1.1	Dictionary of ISIS neighbors keyed by SystemID.

2.801.1.1.1 IsisSystemIdNeighbors

adjacencies	*list of IsisNeighborEntry* see 2.801.1.1.1.1	List of neighbors for SystemID.

2.801.1.1.1.1 IsisNeighborEntry

circuitId	*string*	Circuit Identifier.
[details]	*IsisNeighborDetails* see 2.801.1.1.1.2	ISIS neighbor details.
[hostname]	*string*	Hostname.
interfaceName	*interface name*	ISIS Interface Name.
lastHelloTime	*integer*	UTC timestamp when the last hello was received.
level	*enum: level-1-2 | level-2 | level-1*	Level information of the neighbor.
[routerIdV4] **	*IPv4 address*	ISIS neighbor router ID in IP format. **Since: 4.22.0
[snpa]	*string*	Sub Network Point Of Attachment Address.
state	*enum: down | up | init*	ISIS Adjacency state.

2.801.1.1.1.2 IsisNeighborDetails

advertisedHoldTime	*integer*	Negotiated hold time (in seconds) between neighbors.
areaIds	*list of string*	ISIS Neighbor area(s).

[bfdIpv4State]	*enum: down* | *adminDown* | *NA* | *init* | *up*	ISIS neighbor BFD state. adminDown - Session taken down for administrative purposes init - Waiting for session establishment up - Session established successfully down - Session is down or has just been created.
[bfdIpv6State]	*enum: down* | *adminDown* | *NA* | *init* | *up*	ISIS neighbor BFD state. adminDown - Session taken down for administrative purposes init - Waiting for session establishment up - Session established successfully down - Session is down or has just been created.
[grState]	*string*	Shows the state of GR.
grSupported	*string*	Shows whether GR is Supported or Not.
interfaceAddressFamily **	*enum: ipv4* | *none* | *both* | *ipv6*	Address Family configured on this interface none : Address Family not configured on interface ipv4 : IPv4 configured on interface ipv6 : IPv6 configured on interface both : IPv4 and IPv6 configured on interface. **Since: 4.26.0
[ip4Address]	*IPv4 address*	ISIS Neighbor IPv4 address.
[ip6Address]	*IPv6 address*	ISIS Neighbor IPv6 address.
[ip6GlobalAddress] **	*IPv6 address*	ISIS Neighbor Global IPv6 address. **Since: 4.22.0
neighborAddressFamily **	*enum: ipv4* | *none* | *both* | *ipv6*	Address Family advertised by neighbor none : Address Family not advertised by neighbor ipv4 : IPv4 advertised by neighbor ipv6 : IPv6 advertised by neighbor both : IPv4 and IPv6 advertised by neighbor. **Since: 4.26.0
[priority]	*integer*	ISIS neighbor priority.
srEnabled	*boolean*	Shows whether SR enabled on dut.
[srInfoDetail]	*SrInfoDetail* see 2.801.1.1.1.3	Information about segment routing.
stateChanged	*integer*	UTC Time at which the state changed.

2.801.1.1.1.3 SrInfoDetail

srGbBase	*integer*	SR Base Value.
srGbRange	*integer*	SR Range.
[srLabelV4]	*integer*	Adjacency Label for V4.
[srLabelV6]	*integer*	Adjacency Label for V6.
srRouterId	*IPv4 address*	SR Router Id.

2.802 show isis segment-routing tunnel [...]

revision 2

entries	*table(numeric string) = IsisSrTunnelTableEntry* see 2.802.1	Tunnel table entries keyed by tunnel index.

2.802.1 IsisSrTunnelTableEntry

endpoint	*IP address prefix*	Endpoint of the tunnel.
vias	*list of Via* see 2.151.1.1	List of nexthops.

2.803 show isis summary | (INSTANCE summary) | (summary vrf VRF-NAME)

vrfs	*table(string) = Instances-IsisInstanceSummaryModel* see 2.803.1	ISIS instance information for all VRFs.

2.803.1 InstancesIsisInstanceSummaryModel

isisInstances	*table(string) = IsisInstanceSummaryModel* see 2.803.1.1	IS-IS Instance list keyed by instance name.

2.803.1.1 IsisInstanceSummaryModel

addressFamily	*enum: ipv4 | none | both | ipv6*	Address Family none : Address Family disabled ipv4 : Routes ipv4 only ipv6 : Routes ipv6 only both : Routes both ipv4 and ipv6.
areaIds	*list of string*	List of area-ids in this instance.
[areaLeaderL1] **	*string*	Area leader for Level 1. **Since: 4.23.0
[areaLeaderL2] **	*string*	Area leader for Level 2. **Since: 4.23.0
attached	*boolean*	IPv4 Attached or not.
[authModeKeyIdL1] **	*integer*	Authentication mode key id in Level1. **Since: 4.21.3
[authModeKeyIdL2] **	*integer*	Authentication mode key id in Level2. **Since: 4.21.3
[authModeL1]	*enum: md5 | none | sha | text*	Authentication mode Level1 none : No Authentication md5 : Authentication mode is MD5 text : Authentication mode is TEXT.
[authModeL2]	*enum: md5 | none | sha | text*	Authentication mode Level2 none : No Authentication md5 : Authentication mode is MD5 text : Authentication mode is TEXT.
[authRxDisabledL1] **	*boolean*	Received LSP authentication check is disabled for level-1. **Since: 4.22.1
[authRxDisabledL2] **	*boolean*	Received LSP authentication check is disabled for level-2. **Since: 4.22.1
csnpGenInterval **	*integer*	CSNP generation interval. **Since: 4.27.0
dynamicFloodingL1 **	*boolean*	Dynamic Flooding is enabled for Level 1. **Since: 4.23.0
dynamicFloodingL2 **	*boolean*	Dynamic Flooding is enabled for Level 2. **Since: 4.23.0
enabled	*boolean*	Instance Enabled or not.
gracefulRestart	*boolean*	GR Enabled or not.
gracefulRestartHelper	*boolean*	GR helper Enabled or not.

[hostName]	*string*	Hostname.
instanceId **	*integer*	Instance ID. **Since: 4.24.0
isType	*enum: level2 | level1 | level1-2 | unknown*	Type of IS(Intermediate System) unknown : unknown IS Type level1 : IS type is level-1 level2 : IS type is level-2 level1-2 : IS type is level-1-2.
[lsdbSizeL1]	*integer*	LSDB size (number of link state database records)for level1.
[lsdbSizeL2]	*integer*	LSDB size (number of link state database records)for level2.
lspGenInfo	*IsisInstanceSummaryLsp-GenInterval* see 2.803.1.1.1	Lsp Gen Information.
lspSizeMaxL1 **	*integer*	LSP size maximum for Level 1. **Since: 4.24.0
lspSizeMaxL2 **	*integer*	LSP size maximum for Level 2. **Since: 4.24.0
[msdBaseMplsImposition] **	*integer*	MSD type base MPLS imposition (number of MPLS label which can be imposed). **Since: 4.28.1
multiTopologyAttached	*boolean*	IPv6 Attached or not.
multiTopologyEnabled	*boolean*	multiTopology Enabled or not.
numActiveInterfaces	*integer*	Number of active interfaces.
[numDisInterfacesL1]	*integer*	Number of DIS interfaces for leve1.
[numDisInterfacesL2]	*integer*	Number of DIS interfaces for level2.
overloadBitSetL1 **	*boolean*	Overload bit set for Level 1 LSP. **Since: 4.26.2
overloadBitSetL2 **	*boolean*	Overload bit set for Level 2 LSP. **Since: 4.26.2
[overloadBitUpdate-TimestampL1] **	*floating point number*	UTC timestamp when overload bit status for Level 1 LSP was updated. **Since: 4.26.2
[overloadBitUpdate-TimestampL2] **	*floating point number*	UTC timestamp when overload bit status for Level 2 LSP was updated. **Since: 4.26.2
preferenceInfo	*IsisInstanceSummary-PreferenceModel* see 2.803.1.1.2	Preference Information.
redistRoutesCountL1 **	*integer*	Redistributed routes count for Level 1. **Since: 4.26.2
redistRoutesCountL2 **	*integer*	Redistributed routes count for Level 2. **Since: 4.26.2
redistRoutesMaxLimitL1 **	*integer*	Maximum redistributed routes allowed for Level 1. **Since: 4.26.2
redistRoutesMaxLimitL2 **	*integer*	Maximum redistributed routes allowed for Level 2. **Since: 4.26.2
[routerIdV4] **	*IPv4 address*	ISIS router ID in IP format. **Since: 4.22.0

[sharedSecretProfileL1] **	*string*	Shared secret profile for Level-1. **Since: 4.30.0
[sharedSecretProfileL2] **	*string*	Shared secret profile for Level-2. **Since: 4.30.0
[shortcutSpfForIgp] **	*boolean*	IPv4 Shortcut SPF enabled for IGP. **Since: 4.25.0
[shortcutSpfForLdp-Tunneling] **	*boolean*	Shortcut SPF enabled for LDP tunneling. **Since: 4.25.0
spfInfo	*IsisInstanceSummarySpf-Model* see 2.803.1.1.3	Spf Information.
systemId	*string*	System Id.
v6ShortcutSpfForIgp **	*boolean*	IPv6 Shortcut SPF enabled for IGP. **Since: 4.26.2

2.803.1.1.1 IsisInstanceSummaryLspGenInterval

lspGenHoldInterval	*integer*	Lsp Generation Interval:Hold Interval(s).
lspGenInitialWaitInterval	*integer*	Lsp Generation Interval:Initial wait(ms).
lspGenMaxWaitInterval	*integer*	Lsp Generation Interval:Max wait(s).

2.803.1.1.2 IsisInstanceSummaryPreferenceModel

[internalV4PreferenceL1]	*integer*	internal preference of level1.
[internalV4PreferenceL2]	*integer*	internal preference of level2.
[internalV6PreferenceL1]	*integer*	internal preference of level1.
[internalV6PreferenceL2]	*integer*	internal preference of level2.

2.803.1.1.3 IsisInstanceSummarySpfModel

[spfCurrentHoldIntervalL1]	*integer*	Current SPF hold interval(ms) for Level1.
[spfCurrentHoldIntervalL2]	*integer*	Current SPF hold interval(ms) for Level2.
spfHoldInterval	*integer*	Spf Interval:Hold Interval(ms).
spfInitialWaitInterval	*integer*	Spf Interval:Initial wait(ms).
[spfLastRunL1]	*floating point number*	Last Level 1 SPF ran time.
[spfLastRunL2]	*floating point number*	Last Level 2 SPF ran time.
spfMaxWaitInterval	*integer*	Spf Interval:Max wait(s).

2.804 show isis ti-lfa path [...]

vrfs	*table(string) = TilfaPathVrfModel* see 2.804.1	A mapping between a vrf and its TI-LFA paths.

2.804.1 TilfaPathVrfModel

[v4Info]	*TilfaPathAfModel* see 2.804.1.1	TI-LFA path information for IPv4 address family.
[v6Info]	*TilfaPathAfModel* see 2.804.1.1	TI-LFA path information for IPv6 address family.

2.804.1.1 TilfaPathAfModel

topologies	*table(numeric string) = TilfaPathTopoIdModel* see 2.804.1.1.1	A mapping of topology ID to all TI-LFA paths.

2.804.1.1.1 TilfaPathTopoIdModel

destinations	*table(string) = TilfaPathDestModel* see 2.804.1.1.1.1	A mapping of path systemIds to all TI-LFA paths for that destination.

2.804.1.1.1.1 TilfaPathDestModel

[hostname]	*string*	Hostname.
pathIds	*table(numeric string) = TilfaPathEntryModel* see 2.804.1.1.1.2	A mapping of an ID representing a constraint to the corresponding TI-LFA path.

2.804.1.1.1.2 TilfaPathEntryModel

constraint	*TilfaPathConstraintModel* see 2.804.1.1.1.4	Path constraint.
[details]	*TilfaPathDetails* see 2.804.1.1.1.7	Detailed path information.
status	*enum: pathNotFound | cspfPending | pathFound*	Status of path computation.
sysIds	*list of SystemIdHostnameModel* see 2.804.1.1.1.3	Ordered list of system identifiers or hostname specifying the path to the destination.

2.804.1.1.1.3 SystemIdHostnameModel

[hostname]	*string*	Hostname.
sysId	*string*	System identifier.

2.804.1.1.1.4 TilfaPathConstraintModel

algoId **	*integer*	Algorithm Id. **Since: 4.30.1
[algoName] **	*string*	Algorithm name. **Since: 4.30.1
[excludeIntf]	*interface name*	Interface excluded from this path.
[excludeNode]	*string*	Node excluded from this path.
[excludeSrlgMode] **	*enum: loose | strict*	Type of SRLG exclusion. **Since: 4.24.2
[excludeSrlgOfIntf] **	*IntfSrlg* see 2.804.1.1.1.6	SRLG exclude of interface. **Since: 4.24.2
[flexAlgo] **	*FlexAlgoPathConstraint-Model* see 2.804.1.1.1.5	FlexAlgo constraint. **Since: 4.30.1

2.804.1.1.1.5 FlexAlgoPathConstraintModel

[excludeAdminGroup]	*integer*	Admin group excluded.
[excludeSrlg]	*list of integer*	List of SRLG IDs excluded.
[excludeSrlgMode]	*enum: loose | strict*	Type of SRLG exclusion.
[includeAllAdminGroup]	*integer*	Admin group included.
[includeAnyAdminGroup]	*integer*	Any admin group included.
metricType	*enum: igp | minDelay | te*	Metric type.

2.804.1.1.1.6 IntfSrlg

interface	*interface name*	Interface name.
[srlgIds]	*list of integer*	List of SRLG IDs of the interface.

2.804.1.1.1.7 TilfaPathDetails

[changeCount]	*integer*	Number of times path updated.
[lastUpdatedTime]	*floating point number*	UTC timestamp of the last path update.
refreshReqSeq	*integer*	Request sequence number.
[refreshRespSeq]	*integer*	Response sequence number.

2.805 show isis ti-lfa tunnel [...]

entries	*table(numeric string) = TilfaTunnelTableEntry* see 2.805.1	Tunnel table entries keyed by tunnel index.

2.805.1 TilfaTunnelTableEntry

backupVias	*list of MplsVia* see 2.151.1.3	List of tunnel backup vias.
tunnelId	*integer*	Tunnel identifier.

tunnelIndex	*integer*	Tunnel index within tunnel type.
tunnelType	*enum: RSVP* | *VXLAN VTEP Encap* | *Nexthop* | *IS-IS SR* | *EosSdk sBFD* | *Static MPLS Multicast* | *IS-IS FlexAlgo* | *Nexthop Group* | *Static MPLS* | *VXLAN* | *maxTunnelType* | *RSVP FRR* | *RSVP LER* | *SR-TE* | *invalidTunnel* | *SR-TE Policy* | *BGP LU* | *EVPN VXLAN* | *MLDP P2MP* | *LDP* | *Static Interface* | *IP-in-IP* | *TI-LFA* | *SRv6 Transport* | *BGP LU Forwarding* | *RSVP LER SUB*	Tunnel type.
vias	*list of MplsVia* see 2.151.1.3	List of tunnel vias.

2.806 show isis topology | (INSTANCE topology) | (topology vrf VRF-NAME) [...]

see 2.788

2.807 show kernel interface addr [...]

vrfs	*table(string) = ShowKernelInterfaceAddrVrf* see 2.807.1	A mapping of VRF to its interfaces.

2.807.1 ShowKernelInterfaceAddrVrf

interfaces	*table(string) = ShowKernelInterfaceAddrDetails* see 2.807.1.1	A mapping of interface to its details.

2.807.1.1 ShowKernelInterfaceAddrDetails

[brd]	*string*	Broadcast address.
[extraDetails]	*table(string) = string*	A mapping of extra kernel details to their values.
flags	*string*	Interface flags.
[group]	*string*	Group associated with interface.
inet	*table(string) = ShowKernelInterfaceAddrInet* see 2.807.1.1.1	A mapping of IPv4 internet networking scope to its details.
inet6	*table(string) = ShowKernelInterfaceAddrInet* see 2.807.1.1.1	A mapping of IPv6 internet networking scope to its details.
[link]	*string*	Link address.
[linkNetnsid]	*integer*	Link network namespace id.
mtu	*integer*	Maximum transmission unit.
[qdisc]	*string*	Queuing discipline.
[qlen]	*integer*	Queue length.
[state]	*enum: notPresent | lowerLayerDown | unknown | testing | down | dormant | up*	Interface state.

2.807.1.1.1 ShowKernelInterfaceAddrInet

[addr]	*IP address*	Inet(6) address.
[brd]	*IP address*	Broadcast address.
preferredLft	*string*	Preferred lifetime.
validLft	*string*	Valid lifetime.

2.808 show kernel interface counters [...]

vrfs	*table(string) = ShowKernelInterfaceCountersVrf* see 2.808.1	A mapping of VRFs to interfaces.

2.808.1 ShowKernelInterfaceCountersVrf

interfaces	*table(string) = ShowKernelInterfaceCountersVrfDetails* see 2.808.1.1	A mapping of interfaces to interface details.

2.808.1.1 ShowKernelInterfaceCountersVrfDetails

[broadcast]	*IPv4 address*	Broadcast address.
[deviceInterrupt]	*integer*	Device interrupt number.
[ether]	*MAC address*	Ethernet(MAC) address.
flags	*string*	Flag information.
[inet]	*IPv4 address*	IPv4 internet networking.
[inet6]	*table(string) = ShowKernelInterfaceCountersInet6* see 2.808.1.1.1	A mapping of IPv6 internet networking to its details.
[loop]	*string*	Type of loop.
mtu	*integer*	Maximum transmission unit.
[netmask]	*IPv4 address*	Netmask mask.
rx	*ShowKernelInterfaceCountersRxTxDetails* see 2.808.1.1.2	Packets received details.
tx	*ShowKernelInterfaceCountersRxTxDetails* see 2.808.1.1.2	Packets transmitted details.
txqueuelen	*integer*	Transaction queue length.
[unspec]	*string*	Unspecified address.

2.808.1.1.1 ShowKernelInterfaceCountersInet6

address	*IPv6 address*	IPv6 address.
[prefixlen]	*integer*	Prefix length.
[scopeid]	*integer*	Scope ID.

2.808.1.1.2 ShowKernelInterfaceCountersRxTxDetails

[carrier]	*integer*	Number of carriers.
[collisions]	*integer*	Number of collisions.
dropped	*integer*	Number of frames dropped.
[misaligned]	*integer*	Number of misaligned frames.
numBytes	*integer*	Number of bytes.
numErrors	*integer*	Number of errors.
numPackets	*integer*	Number of packets.

`overruns`	*integer*	Number of overruns.

2.809 show kernel ip acl [...]

vrfs	*table(string) = ShowKernelIpAclVrf* see 2.809.1	A mapping of VRFs to its ACL.

2.809.1 ShowKernelIpAclVrf

chainAcls	*table(string) = ShowKernelIpAclChain* see 2.809.1.1	A mapping of ACLs to their details.

2.809.1.1 ShowKernelIpAclChain

[chainBytes]	*integer*	Number of bytes in ACL.
[chainPackets]	*integer*	Number of packets in ACL.
details	*list of ShowKernelIpAclDetails* see 2.809.1.1.1	A list of the ACLs details.
[policy]	*string*	Policy type.
[references]	*integer*	Number of references.

2.809.1.1.1 ShowKernelIpAclDetails

[aclExtraDetails]	*string*	ACL extra details.
destination	*AclDetailsInvertedIp-Address* see 2.809.1.1.1.1	Destination address.
inbound	*AclDetailsInvertedString* see 2.809.1.1.1.2	Inbound rule.
numBytes	*integer*	Number of bytes.
numPackets	*integer*	Number of packets.
opt	*AclDetailsInvertedString* see 2.809.1.1.1.2	Interface option.
outbound	*AclDetailsInvertedString* see 2.809.1.1.1.2	Outbound rule.
proto	*AclDetailsInvertedString* see 2.809.1.1.1.2	IP protocol.
source	*AclDetailsInvertedIp-Address* see 2.809.1.1.1.1	Source address.
target	*string*	Target value.

2.809.1.1.1.1 AclDetailsInvertedIpAddress

inverted	*boolean*	Attribute is inverted.
value	*IP address*	ACL details value.

2.809.1.1.1.2 AclDetailsInvertedString

inverted	*boolean*	Attribute is inverted.
value	*string*	ACL details value.

2.810 show kernel ip arp [...]

vrfs	*table(string) = ShowKernelIpArpVrf* see 2.810.1	A mapping of VRFs to its ARP tables.

2.810.1 ShowKernelIpArpVrf

arpAddresses	*table(IPv4 or IPv6 address) = ShowKernelIpArpDetails* see 2.810.1.1	A mapping of ARP addresses to their details.

2.810.1.1 ShowKernelIpArpDetails

dev	*string*	ARP device name.
[externallyLearned]	*boolean*	This entry was externally learned.
[lladdr]	*MAC address*	Link layer address.
[proxy]	*boolean*	This neighbor is being proxied.
[router]	*boolean*	This neighbor is a router.
state	*enum: stale | noArp | failed | permanent | reachable | probe | none | delay | incomplete*	State of neighbour entry.

2.811 show kernel ip counters [...]

vrfs	*table(string) = ShowKernelIpCountersVrfDetails* see 2.811.1	A mapping of VRFs to detail sections.

2.811.1 ShowKernelIpCountersVrfDetails

arista	*table(string) = integer*	A mapping of Arista details to Arista values.
[icmp]	*ShowKernelIpCountersIcmp* see 2.811.1.1	VRF ICMP infomation.
[icmp6]	*ShowKernelIpCountersIcmp* see 2.811.1.1	VRF ICMP6 information.
icmpMsg	*table(string) = integer*	A mapping of ICMPMsg details to IMCPMsg values.
[ip]	*table(string) = integer*	A mapping of IP details to IP values.
[ip6]	*table(string) = integer*	A mapping IP6 details to IP6 values.
ipExt	*table(string) = integer*	A mapping of IP extension details to IP extension values.
[sctp]	*table(string) = integer*	A mapping of SCTP details to SCTP values.
tcp	*table(string) = integer*	A mapping of TCP details to TCP values.
tcpExt	*table(string) = integer*	A mapping of TCP extension details to TCP extension values.
[udp]	*table(string) = integer*	A mapping of UDP details to UDP values.
[udp6]	*table(string) = integer*	A mapping of UDP6 details to UDP6 values.
[udpLite]	*table(string) = integer*	A mapping of UDPLite details to UDPLite values.
[udpLite6]	*table(string) = integer*	A mapping of UDPLite6 details to UDPLite6 values.

2.811.1.1 ShowKernelIpCountersIcmp

details	*table(string) = integer*	A mapping of ICMP(6) details to ICMP(6) values.
inputHistogram	*table(string) = integer*	A mapping of ICMP(6) input histogram details to input histogram values.
outputHistogram	*table(string) = integer*	A mapping of ICMP(6) output histogram details to output histogram values.

2.812 show kernel ip interface addr [...]

see 2.807

2.813 show kernel ip route [...]

vrfs	*table(string) = ShowKernelIpRouteVrf* see 2.813.1	A mapping of VRFs to their IP routes.

2.813.1 ShowKernelIpRouteVrf

routes	*table(IPv4 or IPv6 address) = ShowKernelIpRouteDetails* see 2.813.1.1	A mapping of IP routes to route details.

2.813.1.1 ShowKernelIpRouteDetails

[dead]	*boolean*	Route has dead flag.
[dev]	*string*	Device name.
[errorNum]	*integer*	Error number.
[linkdown]	*boolean*	Route has linkdown flag.
[metric]	*integer*	Metric value.
[notify]	*boolean*	Route has notify flag.
[offload]	*boolean*	Route has offload flag.
[onlink]	*boolean*	Route has onlink flag.
[pervasive]	*boolean*	Route has pervasive flag.
[pref]	*enum: high | medium | low*	Preference value.
[proto]	*string*	Protocol type.
[routeType]	*string*	Route type.
[scope]	*string*	Route scope.
[src]	*IP address*	Source address.
[unresolved]	*boolean*	Route has unresolved flag.
[via]	*IP address*	Next hop address.

2.814 show kernel ipsec [...]

[connections]	*table(string) = ShowKernelIpsecConnections* see 2.814.1	IPsec Connections.
[dataPath]	*table(string) = integer*	Details of /proc/net/xfrm_stat.
[listeningIpAddresses]	*list of IP address*	IPsec Listening IP Addresses.
[securityAssociations]	*table(string) = ShowKernelIpsecSecurityAssociations* see 2.814.2	IPsec Security Associations.
[strongSwan]	*ShowKernelIpsecStrongSwan* see 2.814.3	Details about IKE charon daemon.

2.814.1 ShowKernelIpsecConnections

summary	*list of string*	Details about a connection.

2.814.2 ShowKernelIpsecSecurityAssociations

[ikeSa]	*table(string) = ShowKernelIpsecIkeSa* see 2.814.2.1	Details about each security associations ikeSa's and related childSa's.

2.814.2.1 ShowKernelIpsecIkeSa

childSa	*table(string) = ShowKernelIpsecChildSa* see 2.814.2.1.1	ChildSa's relevant to this ikeSa.
summary	*list of string*	Details about an ikeSa.

2.814.2.1.1 ShowKernelIpsecChildSa

summary	*list of string*	Details about a childSa.

2.814.3 ShowKernelIpsecStrongSwan

architecture	*string*	Architecture running on.
linux	*string*	Linux kernel version.
malloc	*table(string) = integer*	Details of memory.
[plugins]	*list of string*	Plugins enabled.
startTime	*floating point number*	Start time in UTC.
uptime	*integer*	Uptime in seconds.
version	*string*	StrongSwan version number.
workerThreads	*ShowKernelIpsecWorkerThreads* see 2.814.3.1	Details of all worker threads.

2.814.3.1 ShowKernelIpsecWorkerThreads

idle	*integer*	Number of idle threads.
queue	*string*	Number of threads in the queue.
scheduled	*integer*	Number of scheduled threads.
total	*integer*	Total number of threads.
working	*string*	Number of working threads.

2.815 show kernel ipv6 acl [...]

see 2.809

2.816 show kernel ipv6 counters [...]

see 2.811

2.817 show kernel ipv6 interface addr [...]

see 2.807

2.818 show kernel ipv6 neighbors [...]

see 2.810

2.819 show kernel ipv6 route [...]

see 2.813

2.820 show kernelFib rcf [...]

see 2.194

2.821 show kernelFib scheduler [...]

see 2.79

2.822 show l1 destination [...]

revision 2

interfaces	*table(Interface name) = L1Destination* see 2.822.1	A mapping of source interface to a list of destination interfaces and connection type.
physicalInterfaces	*table(Interface name) = VirtualIntfs* see 2.822.3	A mapping of physical interface name to virtual interface name.
virtualInterfaces	*table(Interface name) = PhysicalIntfs* see 2.822.2	A mapping of virtual interface name to physical interface name.

2.822.1 L1Destination

[connectionType]	*enum: dynamic | static*	L1 Connection Type.
destinationInterfaces	*list of string*	A list of destination interfaces.

2.822.2 PhysicalIntfs

[rxPhysicalInterface]	*interface name*	Rx physical interface.
[txPhysicalInterface]	*interface name*	Tx physical interface.

2.822.3 VirtualIntfs

[rxVirtualInterfaces]	*list of Interface name*	A list of Rx virtual interfaces.
[txVirtualInterface]	*interface name*	Tx virtual Interface.

2.823 show l1 matrix [...]

revision 2

interfaces	*table(string) = L1MatrixEntry* see 2.823.1	A mapping of interface to its source and destinations.

2.823.1 L1MatrixEntry

[description]	*string*	Interface description.
destinations	*list of string*	A list of destination interfaces.
[source]	*string*	Interface name.

2.824 show l1 modules profile configuration [...]

[builtin]	*table(string) = L1CardProfileEnumeration* see 2.824.1	The builtin L1 module profiles installed on the system.

2.824.1 L1CardProfileEnumeration

[applicableModules]	*list of string*	Applicable modules of L1 module profile.
description	*string*	The description of the L1 module profile.

2.825 show l1 modules profile status [...]

modules	*table(string) = L1ProfileCardStatus* see 2.825.1	The L1 module profile configuration status for a particular module.

2.825.1 L1ProfileCardStatus

[configured]	*string*	The configured L1 module profile.
[operational]	*string*	The operational L1 module profile.
status	*enum: notInserted | error | pending | applied*	The status of the L1 module profile configuration.

2.826 show l1 path [...]

revision 2

paths	*table(Interface name) = L1Source* see 2.826.1	A list of paths.
physicalInterfaces	*table(Interface name) = VirtualIntfs* see 2.822.3	A mapping of physical interface name to virtual interface name.
virtualInterfaces	*table(Interface name) = PhysicalIntfs* see 2.822.2	A mapping of virtual interface name to physical interface name.

2.826.1 L1Source

[connectionType]	*enum: dynamic | static*	L1 Connection Type.
[sourceInterface]	*string*	Interface Name.

2.827 show l1 source [...]

revision 2

interfaces	*table(Interface name) = L1Source* see 2.826.1	A mapping of destination interface to source interface and connection type.
physicalInterfaces	*table(Interface name) = VirtualIntfs* see 2.822.3	A mapping of physical interface name to virtual interface name.
virtualInterfaces	*table(Interface name) = PhysicalIntfs* see 2.822.2	A mapping of virtual interface name to physical interface name.

2.828 show l1 source capabilities [...]

interfaces	*table(Interface name) = SourceableIntfs* see 2.828.1	A mapping of interface to sourceable interfaces.

2.828.1 SourceableIntfs

sources	*list of Interface name*	A list of sourceable interfaces.

2.829 show l2-protocol forwarding interfaces [...]

interfaces	*table(Interface name) = L2ProtocolForwarding-Interface* see 2.829.1	L2 protocol forwarding information for interfaces.

2.829.1 L2ProtocolForwardingInterface

profileName	*string*	Active L2 protocol forwarding profile.
[protocolInfo]	*list of L2ProtocolForwardingInfo* see 2.829.1.1	L2 protocol forwarding information.

2.829.1.1 L2ProtocolForwardingInfo

action	*enum: drop | forwardCpu | forward*	Action to be performed on protocol packets.
[protocol]	*enum: mbfd | lldp | lacp | pause | macsec | isis | elmi | stp*	L2 Protocol.
[tagFormat] **	*enum: all | untagged | tagged*	VLAN tag format. **Since: 4.21.3

2.830 show l2-protocol forwarding profile [...]

profiles	*table(string) = L2Protocol-ForwardingProfile* see 2.830.1	A mapping of L2 protocol forwarding profile name to profile information.

2.830.1 L2ProtocolForwardingProfile

[protocolInfo]	*list of L2ProtocolForwardingInfo* see 2.829.1.1	L2 protocol forwarding information.

2.831 show l2-protocol forwarding profile summary [...]

profiles	*table(string) = L2Protocol-ForwardingProfileSummary* see 2.831.1	A mapping of L2 protocol forwarding profile name to interfaces on which it is active/configured.

2.831.1 L2ProtocolForwardingProfileSummary

[activeIntfs]	*list of Interface name*	Interfaces on which profile is active.
[configuredIntfs]	*list of Interface name*	Interfaces on which profile is configured.

2.832 show l2Rib (input SOURCE) [...]

hosts	*list of HostEntry* see 2.832.1	Host table entries.
invalidHosts	*list of HostEntry* see 2.832.1	Invalid host table entries.

2.832.1 HostEntry

Could also be an instance of:
HostEntryAndSource 2.832.2

dests	*list of L2RibObject* see 2.832.1.1	collection of destination objects.
entryType	*string*	Host entry type.
macAddress	*MAC address*	MAC Address.
pref	*integer*	Preference for the host entry.
seqNo	*integer*	Sequence number.
vlanId	*integer*	VLAN Id of MAC address.

2.832.1.1 L2RibObject

Could also be an instance of:
Dest 2.832.1.2
Label 2.832.1.3
LoadBalance 2.832.1.4

[index]	*integer*	Object table key.
level	*integer*	Level in the chain of objects.
spacePerLevel	*integer*	Number of spaces to indent per level.

2.832.1.2 Dest

[color] **	*integer*	Color of SR-TE policy tunnel. **Since: 4.24.1
destType	*enum: MPLS | VXLAN | CPU | Tunnel | Interface*	Destination type.
[index]	*integer*	Object table key.
[interface]	*interface name*	Local interface.
level	*integer*	Level in the chain of objects.
[mplsLabel]	*integer*	MPLS Label.
spacePerLevel	*integer*	Number of spaces to indent per level.
[tunnelEndPoint] **	*IP address prefix*	IP prefix of the MPLS tunnel endpoint. **Since: 4.20.5
[tunnelId]	*TunnelId* see 2.151.1.3.1.4	Tunnel Identifier.
[vtepAddr]	*IpGenericAddr* see 2.296.1.1.1.1	IP address of VXLAN tunnel endpoint.

2.832.1.3 Label

[index]	*integer*	Object table key.
label	*integer*	MPLS Label.

level	*integer*	Level in the chain of objects.
spacePerLevel	*integer*	Number of spaces to indent per level.

2.832.1.4 LoadBalance

[index]	*integer*	Object table key.
level	*integer*	Level in the chain of objects.
num	*integer*	Number of load balance objects.
spacePerLevel	*integer*	Number of spaces to indent per level.

2.832.2 HostEntryAndSource

dests	*list of L2RibObject* see 2.832.1.1	collection of destination objects.
entryType	*string*	Host entry type.
macAddress	*MAC address*	MAC Address.
pref	*integer*	Preference for the host entry.
seqNo	*integer*	Sequence number.
source	*string*	host entry source.
vlanId	*integer*	VLAN Id of MAC address.

2.833 show l2Rib (input SOURCE) floodset [...]

floodSetSummaries	*list of FloodSetSummary* see 2.833.1	Vlan flood set summary collection.

2.833.1 FloodSetSummary

floodSets	*list of FloodSet* see 2.833.1.1	Vlan flood set summary.
[source]	*string*	L2Rib Flood Set table source.
tableName	*string*	L2Rib Flood Set table name.

2.833.1.1 FloodSet

dests	*list of Dest* see 2.832.1.2	Flood destination.
floodType	*enum: Any | All*	Flood type.
macAddr	*MAC address*	Flood set MAC address.
vlanId	*integer*	Vlan Id.

2.834 show l2Rib output [...]

see 2.832

2.835 show l2Rib output floodset [...]

see 2.833

2.836 show l2Rib summary

tables	*list of TableSummary* see 2.836.1	Number of entries in each table.

2.836.1 TableSummary

hosts	*integer*	Number of hosts in table.
name	*string*	Table name.

2.837 show lacp aggregates [...]

[markerMessages]	*LacpMarkerMessages* see 2.837.1	LACP annotation markers and messages.
otherAggregates	*table(string) = PortChannelAggregate* see 2.837.2	Aggregates not bound to any port-channel.
portChannels	*table(Interface name) = PortChannelAggregate* see 2.837.2	Map port-channel to aggregate IDs and list of bundled ports.

2.837.1 LacpMarkerMessages

markerMessages	*list of LacpMarkerMessage* see 2.837.1.1	LACP annotation markers and messages.

2.837.1.1 LacpMarkerMessage

marker	*string*	LACP annotation marker.
msg	*string*	LACP annotation marker message.

2.837.2 PortChannelAggregate

[aggregateId]	*string*	Aggregate ID.
bundledPorts	*list of Interface name*	Ports in aggregate.
incompatiblePortsDetail	*table(string) = IncompatiblePorts* see 2.837.2.1	Ports aggregable, but in incompatible aggregate indexed by their aggregate ID.
[markers]	*LacpAnnotationMarkerSet* see 2.837.2.2	annotation brief marker.
negotiationPorts	*list of Interface name*	Ports still in negotiation.
otherPorts	*list of Interface name*	Ports that are selected and/or standby but not part of an aggregate.
standbyPorts	*list of Interface name*	Standby Ports.

2.837.2.1 IncompatiblePorts

[ports]	*list of Interface name*	Ports configured to be in port-channel, aggregable, but not in agg.

2.837.2.2 LacpAnnotationMarkerSet

markers	*list of string*	Set of LACP annotation markers.

2.838 show lacp counters [...]

[markerMessages]	*LacpMarkerMessages* see 2.837.1	LACP annotation markers and messages.
orphanPorts	*table(Interface name) = InterfaceLacpCounters* see 2.838.1	LACP counters for orphaned ports.
portChannels	*table(Interface name) = PortChannelLacpCounters* see 2.838.2	Map port-channel LACP traffic statistics.

2.838.1 InterfaceLacpCounters

actorPortStatus	*enum: idle | unknown | mismatchAgg | negotiation | standby | noAgg | unselected | bundled*	LACP port status.
[details]	*InterfaceLacpCounters-Details* see 2.838.1.1	Additional LACP counter information.
illegalRxCount	*integer*	Illegal Rx packet count.
lacpdusRxCount	*integer*	LACP PDUs Rx count.
lacpdusTxCount	*integer*	LACP PDUs Tx count.
markerResponseRxCount	*integer*	Marker response Rx count.
markerResponseTxCount	*integer*	Marker response Tx count.
markersRxCount	*integer*	Markers Rx count.
markersTxCount	*integer*	Markers Tx count.

2.838.1.1 InterfaceLacpCountersDetails

actorChangeCount	*integer*	Actor port change count.
actorChurnCount	*integer*	Actor port churn count.
actorSyncTransitionCount	*integer*	Actor port sync transition count.
lastRxTime	*floating point number*	UTC timestamp of when the actor last received LACP packet.
partnerChangeCount	*integer*	Partner port change count.
partnerChurnCount	*integer*	Partner port churn count.
partnerSyncTransitionCount	*integer*	Partner port sync transition count.

2.838.2 PortChannelLacpCounters

interfaces	*table(Interface name) = InterfaceLacpCounters* see 2.838.1	Map ports to traffic values.
[markers]	*LacpAnnotationMarkerSet* see 2.837.2.2	annotation brief marker.

2.839 show lacp interface [...]

[interface]	*interface name*	LACP information for this interface only.
[markerMessages]	*LacpMarkerMessages* see 2.837.1	LACP annotation markers and messages.
orphanPorts	*table(Interface name) = InterfaceLacp* see 2.839.1	LACP information for orphaned ports.
portChannels	*table(Interface name) = PortChannelLacp* see 2.839.2	Map port-channels to their LACP information.

2.839.1 InterfaceLacp

actorOperKey	*string*	LACP actor operational key.
actorPortId	*integer*	Actor port number.
actorPortPriority	*integer*	Actor port priority.
actorPortState	*LacpPortState* see 2.839.1.2	LACP actor port state.
actorPortStatus	*enum: idle | unknown | mismatchAgg | negotiation | standby | noAgg | unselected | bundled*	LACP port status.
[details]	*InterfaceLacpDetails* see 2.839.1.1	Additional LACP port information.
partnerOperKey	*string*	LACP partner operational key.
partnerPortId	*integer*	Partner port number.
partnerPortPriority	*integer*	Partner port priority.
partnerPortState	*LacpPortState* see 2.839.1.2	LACP partner port state.
partnerSysId	*string*	System Identifier.

2.839.1.1 InterfaceLacpDetails

actorAdminKey	*string*	Actor adminstrative key.
actorChurnState	*enum: churnDetected | churnMonitor | noChurn*	LACP port churn state.
actorLastRxTime	*floating point number*	UTC timestamp of when the actor last received LACP packet.
actorMuxReason	*string*	LACP actor mux reason.
actorMuxSmState	*enum: muxSmDisablingInProgress | muxSmDetached | muxSmWaiting | muxSmCollecting | muxSmAttached | muxSmCollectingDistributing | muxSmDistributing*	LACP MUX state machine state.

actorRxSmState	*enum: rxSmPortDisabled | rxSmExpired | rxSmInitialize | rxSmDefaulted | rxSmLacpDisabled | rxSmCurrent*	LACP Receive state machine state.
[actorTimeoutMultiplier] **	*integer*	LACP actor timeout multiplier. **Since: 4.25.1
aggSelectionState	*enum: selectedStateUnselected | selectedStateSelected | selectedStateStandby*	Interface aggregator selection state.
partnerChurnState	*enum: churnDetected | churnMonitor | noChurn*	LACP port churn state.
partnerCollectorMaxDelay	*integer*	Partner Collector max delay in tens of microseconds.

2.839.1.2 LacpPortState

activity	*boolean*	LACP active or passive.
aggregation	*boolean*	Link aggregateable or individual.
collecting	*boolean*	Collection of incoming frames enabled.
defaulted	*boolean*	Using defaulted operational partner information.
distributing	*boolean*	Distribution of outgoing frames enabled.
expired	*boolean*	Receive machine in expired state.
synchronization	*boolean*	Link in sync with the correct LAG.
timeout	*boolean*	LACP long (False) or short (True) timeout.

2.839.2 PortChannelLacp

interfaces	*table(Interface name) = InterfaceLacp* see 2.839.1	Map ports to port LACP information.
[markers]	*LacpAnnotationMarkerSet* see 2.837.2.2	annotation brief marker.

2.840 show lacp interface counters [...]

see 2.838

2.841 show lacp interface internal [...]

[markerMessages]	*LacpMarkerMessages* see 2.837.1	LACP annotation markers and messages.
[miscLacpSysIds]	*list of LacpSysIdBase* see 2.841.2	Misc System Id.
orphanPorts	*table(Interface name) = InterfaceLacpInternal* see 2.841.1	LACP internal information for orphaned ports.
portChannels	*table(Interface name) = PortChannelLacpInternal* see 2.841.4	Map channel to local LACP state.
[systemId]	*string*	LACP System-identifier.

2.841.1 InterfaceLacpInternal

actorOperKey	*string*	LACP actor operational key used to form aggregate.
actorPortId	*integer*	LACP actor port ID.
actorPortPriority	*integer*	LACP actor port priority.
actorPortState	*LacpPortState* see 2.839.1.2	LACP actor port state.
actorPortStatus	*enum: idle | unknown | mismatchAgg | negotiation | standby | noAgg | unselected | bundled*	LACP port status.
[details]	*InterfaceLacpInternal-Details* see 2.841.1.1	Detailed LACP port information.
partnerSystemId	*string*	LACP partner system identifier.

2.841.1.1 InterfaceLacpInternalDetails

actorAdminKey	*string*	LACP admin key.
actorChurnState	*enum: churnDetected | churnMonitor | noChurn*	LACP port churn state.
actorMuxReason	*string*	LACP actor mux reason.
actorMuxSmState	*enum: muxSmDisablingInProgress | muxSmDetached | muxSmWaiting | muxSmCollecting | muxSmAttached | muxSm-CollectingDistributing | muxSmDistributing*	LACP MUX state machine state.
actorRxSmState	*enum: rxSmPortDisabled | rxSmExpired | rxSmInitialize | rxSmDefaulted | rxSmLacpDisabled | rxSmCurrent*	LACP Receive state machine state.

[actorTimeoutMultiplier] **	*integer*	LACP actor timeout multiplier. **Since: 4.25.1
aggSelectionState	*enum: selectedStateUnselected | selectedStateSelected | selectedStateStandby*	Interface aggregator selection state.
[allowToWarmup]	*boolean*	Is interface allowed to warm up?
lastRxTime	*floating point number*	UTC timestamp of when the actor last received LACP packet.

2.841.2 LacpSysIdBase

Could also be an instance of:
MlagLacpSysId 2.841.3

2.841.3 MlagLacpSysId

priority	*integer*	System priority.
systemId	*string*	System Identifier.

2.841.4 PortChannelLacpInternal

interfaces	*table(Interface name) = InterfaceLacpInternal* see 2.841.1	Map ports to LACP protocol information.
[markers]	*LacpAnnotationMarkerSet* see 2.837.2.2	annotation brief marker.

2.842 show lacp interface neighbor [...]

[markerMessages]	*LacpMarkerMessages* see 2.837.1	LACP annotation markers and messages.
orphanPorts	*table(Interface name) = InterfaceLacpNeighbors* see 2.842.1	LACP neighbor information for orphaned ports.
portChannels	*table(Interface name) = PortChannelLacpNeighbors* see 2.842.2	Map the port-channels to their remote neighbor LACP protocol state.

2.842.1 InterfaceLacpNeighbors

actorPortStatus	*enum: idle | unknown | mismatchAgg | negotiation | standby | noAgg | unselected | bundled*	LACP port status.
[details]	*InterfaceLacpNeighbors-Details* see 2.842.1.1	Detailed LACP port neighbor information.
partnerOperKey	*string*	LACP partner operational key.
partnerPortId	*integer*	LACP partner port ID.
partnerPortPriority	*integer*	LACP partner port priority.
partnerPortState	*LacpPortState* see 2.839.1.2	LACP partner port state.
partnerSystemId	*string*	LACP partner system identifier.

2.842.1.1 InterfaceLacpNeighborsDetails

aggSelectionState	*enum: selectedStateUnselected | selectedStateSelected | selectedStateStandby*	Interface aggregator selection state.
collectorMaxDelay	*integer*	Collector max delay in tens of microseconds.
partnerChurnState	*enum: churnDetected | churnMonitor | noChurn*	LACP port churn state.

2.842.2 PortChannelLacpNeighbors

interfaces	*table(Interface name) = InterfaceLacpNeighbors* see 2.842.1	Map ports to LACP neighbor information.
[markers]	*LacpAnnotationMarkerSet* see 2.837.2.2	annotation brief marker.

2.843 show lacp interface peer [...]

see 2.842

2.844 show lacp internal [...]

see 2.841

2.845 show lacp internal events [...]

lacpInternalEventRecords	*list of LacpInternalEventRecord* see 2.845.1	LACP state transition events.

2.845.1 LacpInternalEventRecord

currentWhileTimerExpired	*boolean*	currentWhile timer has expired.
intfId	*interface name*	Port channel member interface name.
lacpEnabled	*boolean*	LACP is enabled.
lacpduRx	*boolean*	LACPDU is received.
partnerSync	*boolean*	Partner is synced.
portChannel	*interface name*	Port channel name.
portEnabled	*boolean*	Port is enabled.
portMoved	*boolean*	Port has moved.
reason	*string*	Reason for the port channel member state transition.
selected	*enum: selectedStateUnselected* | *selectedStateSelected* | *selectedStateStandby*	Selected state.
smName	*string*	rx or mux state machine.
state	*enum: muxSmDisablingInProgress* | *rxSmPortDisabled* | *muxSmDetached* | *muxSmWaiting* | *muxSmCollecting* | *rxSmExpired* | *muxSmAttached* | *rxSmInitialize* | *rxSmDefaulted* | *muxSmCollectingDistributing* | *muxSmDistributing* | *rxSmLacpDisabled* | *rxSmCurrent*	Port Channel member state.
timestamp	*floating point number*	UTC timestamp of the state transition.

2.846 show lacp neighbor [...]

see 2.842

2.847 show lacp peer [...]

see 2.842

2.848 show lacp sys-id [...]

[details]	*Details* see 2.848.1	LACP System-Id details.
miscLacpSysIds	*list of LacpSysIdBase* see 2.841.2	Misc System-Id.
systemId	*string*	System identifier.

2.848.1 Details

address	*string*	Switch MAC address.
priority	*integer*	System priority.

2.849 show led [...]

systemLeds	*table(string) = LedSetting* see 2.849.1	A mapping of system LED to its settings.

2.849.1 LedSetting

blue	*boolean*	Blue LED is on.
green	*boolean*	Green LED is on.
maxBright	*integer*	Maximum brightness (%).
minBright	*integer*	Minimum brightness (%).
period	*floating point number*	Flash period (seconds).
red	*boolean*	Red LED is on.
status	*enum: inactive | unknown | good | off | warning | beacon | bad*	LED status.
yellow	*boolean*	Yellow LED is on.

2.850 show led palette [...]

ledCapability	*table(string) = LedCapability* see 2.850.1	All LED capabilities; key is name of the LED.

2.850.1 LedCapability

lightSetting	*table(string) = LightSetting* see 2.850.1.1	All LedCapability LightSettings; key is name of the color.
name	*string*	The name of the LED (e.g. Ethernet1/1, Fan1/1, etc.).

2.850.1.1 LightSetting

color	*string*	The color of the LightSetting.
ledFlashRate	*floating point number*	Duration per flash (in seconds) of the LED.An LED flash-rate of 0.0 means no flash.

2.851 show led state [...]

leds	*table(string) = LedStateDict* see 2.851.1	All LEDs with their states, key is the name of the LED.

2.851.1 LedStateDict

name	*string*	The name of the LED.
state	*table(string) = LedState* see 2.851.1.1	All LED states; key is the name of the state.

2.851.1.1 LedState

available	*boolean*	Indicate whether the state is available for the hardware.
name	*string*	The name of the LED state (e.g. linkUpActivity400G, etc.).

2.852 show led theme

theme	*table(string) = LightSettingTheme* see 2.852.1	All themes created, the key is names of the themes.

2.852.1 LightSettingTheme

name	*string*	The name of the theme.
setting	*table(string) = LightSetting* see 2.850.1.1	All theme settings; the key is led state.

2.853 show led theme interface [...]

ledsTheme	*table(string) = string*	Mapping of which LEDs have which themes.
theme	*table(string) = LightSettingTheme* see 2.852.1	All themes created, the key is names of the themes.

2.854 show license [...]

customerName	*string*	Customer name.
deviceSerial	*string*	System Serial number.
domainName	*string*	Domain name.

features	*table(string) = FeatureModel* see 2.854.1	A mapping between feature name and its license information.
[platform] **	*string*	Platform model. **Since: 4.21.0
systemMac	*string*	System MAC address.

2.854.1 FeatureModel

[featureDescription]	*string*	Description of the feature.
licenses	*list of LicenseInstance* see 2.854.1.1	List of licenses for the feature.

2.854.1.1 LicenseInstance

count	*integer*	Number of licenses of this type.
endTime	*floating point number*	POSIX Timestamp of when license will expire.
[parameter]	*string*	Parameter associated with the license.
source **	*string*	Source. **Since: 4.24.0
startTime	*floating point number*	POSIX Timestamp of when license will become valid.

2.855 show license features

active	*list of string*	List of active licenses.
missing	*list of string*	List of missing licenses.

2.856 show license files [...]

licenseText	*table(string) = string*	A mapping between license file name and its contents.

2.857 show line system dom [...]

see 2.504

2.858 show line system status [...]

ports	*table(string) = InterfacesTransceiverStatusValues* see 2.858.1	Mapping between port name and transceiver status.

2.858.1 InterfacesTransceiverStatusValues

[adapters]	*InterfacesTransceiverStatuValuesSwizzlerList* see 2.858.1.1	Form factor adapter types in use.
[badEepromChecksums]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Number of bad Eeprom checksums.
[cfp2Dco]	*InterfacesTransceiverStatusExtCfp2Dco* see 2.858.1.3	CFP2-DCO specific attributes.
[channels]	*table(numeric string) = InterfacesTransceiverStatusChannelSpecific* see 2.858.1.2	Channel specific attributes indexed by channel number.
[dataPathFirmwareFault]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Data path firmware fault.
[dataPathState]	*table(numeric string) = InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Data path state indexed by host lane number.
[domControlFail]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Number of DOM control/status failures.
[eepromReadTimeouts]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Number of Eeprom read timeouts.
[hostLanes]	*table(numeric string) = InterfacesTransceiverStatusHostLaneSpecific* see 2.858.1.4	Host lane specific attributes indexed by lane number.
[interfaces]	*table(string) = InterfacesTransceiverStatusInterfaceSpecific* see 2.858.1.5	Interface specific attributes indexed by interface name.
[interrupts]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Number of transceiver interrupts.
[mdioFailures]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Number of transceiver management data input/output failures.
mediaType	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Transceiver media type.
[moduleFirmwareFault]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Module firmware fault.

[moduleState]	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state.
[presence]	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Indication of transceiver presence.
[resets]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Number of transceiver resets.
serialNumber	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Transceiver serial number.
[smbusFailures]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Number of transceiver system managements bus failures.
[spuriousDet]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Number of spurious detections of the transceiver.
[tempHiAlarm]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	High temperature alarm.
[tempHiWarn]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	High temperature warn.
[tempLoAlarm]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Low temperature alarm.
[tempLoWarn]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Low temperature warn.
[voltageHiAlarm]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	High voltage alarm.
[voltageHiWarn]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	High voltage alarm.
[voltageLoAlarm]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Low voltage alarm.
[voltageLoWarn]	*InterfacesTransceiver-StatusValueFormat* see 2.858.1.2.3	Low voltage alarm.

2.858.1.1 InterfacesTransceiverStatuValuesSwizzlerList

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*list of string*	List of adapters in use.

2.858.1.2 InterfacesTransceiverStatusChannelSpecific

[coherentAlarms]	*InterfacesTransceiverStatusCoherentChannelSpecific* see 2.858.1.2.1	Coherent media channel specific alarms.
[frequencyTuningAlarms]	*InterfacesTransceiverStatusFrequencyTuningChannelSpecific* see 2.858.1.2.2	Frequency tuning media channel specific CMIS alarms.
[rxCdrLol]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive clock and data recovery loss of lock.
[rxLos]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive loss of signal.
[rxPowerHiAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive high power alarm.
[rxPowerHiWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive high power warning.
[rxPowerLoAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive low power alarm.
[rxPowerLoWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive low power warning.
[txBiasHiAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit high bias alarm.
[txBiasHiWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit high bias warning.
[txBiasLoAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit low bias alarm.
[txBiasLoWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit low bias warning.
[txFault]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit fault.
[txOutputSignalStats]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit output signal status.
[txPowerHiAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit high power alarm.
[txPowerHiWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit high power warning.

[txPowerLoAlarm]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit low power alarm.
[txPowerLoWarn]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit low power warning.

2.858.1.2.1 InterfacesTransceiverStatusCoherentChannelSpecific

[fecDetectedDegrade]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive FEC detect degrade (indicates that BER exceeded the detected degrade threshold).
[fecDetectedDegradeEnabled]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive FEC detected degrade monitoring enabled.
[fecExcessiveDegrade]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive FEC excessive degrade (indicates that BER exceeded the excessive degrade threshold).
[fecExcessiveDegradeEnabled]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive FEC excessive degrade monitoring enabled.
[rxChrDispCompensationLossOfLock]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive chromatic dispersion compensation loss of lock.
[rxDemodulationLossOfLock]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive demodulator loss of lock.
[rxDeskewLossOfLock]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive deskew loss of lock.
[rxFifoErr]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive FIFO error.
[rxLossOfAlignment]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive loss of alignment.
[rxOutOfAlignment]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Receive out of alignment.
[txCmuLossOfLock]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Transmit clock monitor unit clock loss of lock.
[txDeskewLossOfLock]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Transmit deskew clock loss of lock.
[txFifoErr]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Transmit FIFO error.
[txLossOfAlignment]	*InterfacesTransceiverStatusValueFormatBool* see 2.858.1.2.1.1	Transmit loss of alignment.

[txOutOfAlignment]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Transmit out of alignment.
[txRefClkLossOfLock]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Transmit reference clock loss of lock.

2.858.1.2.1.1 InterfacesTransceiverStatusValueFormatBool

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*boolean*	Current value.

2.858.1.2.2 InterfacesTransceiverStatusFrequencyTuningChannelSpecific

[invalidChannel]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Frequency tuning invalid channel (configured channel is outside the range for the selected grid spacing).
[tuningComplete]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Frequency tuning completed.
[tuningInProgress]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Frequency tuning in progress.
[tuningNotAccepted]	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	Frequency tuning busy.

2.858.1.2.3 InterfacesTransceiverStatusValueFormat

Could also be an instance of:
InterfacesTransceiverStatuValuesSwizzlerList 2.858.1.1
InterfacesTransceiverStatusIntfValueFormatEgressAlarmStatus 2.858.1.2.4
InterfacesTransceiverStatusIntfValueFormatIngressAlarmStatus 2.858.1.2.5
InterfacesTransceiverStatusValueFormatBool 2.858.1.2.1.1
InterfacesTransceiverStatusValueFormatFloat 2.858.1.2.6
InterfacesTransceiverStatusValueFormatHex 2.858.1.2.7
InterfacesTransceiverStatusValueFormatInt 2.858.1.2.8
InterfacesTransceiverStatusValueFormatLaneAlignment 2.858.1.2.9
InterfacesTransceiverStatusValueFormatModuleFaultStatus 2.858.1.2.10
InterfacesTransceiverStatusValueFormatModuleGeneralStatus 2.858.1.2.11
InterfacesTransceiverStatusValueFormatModuleStateDec 2.858.1.2.12
InterfacesTransceiverStatusValueFormatNetworkLaneFaultStatus 2.858.1.2.13
InterfacesTransceiverStatusValueFormatNetworkLaneRxOTNStatus 2.858.1.2.14
InterfacesTransceiverStatusValueFormatNetworkLaneVendorFAWS 2.858.1.2.15
InterfacesTransceiverStatusValueFormatRxTurnUpStateDec 2.858.1.2.16
InterfacesTransceiverStatusValueFormatScientificFloat 2.858.1.2.17
InterfacesTransceiverStatusValueFormatStr 2.858.1.2.4.1
InterfacesTransceiverStatusValueFormatTrib 2.858.1.2.18
InterfacesTransceiverStatusValueFormatTxModulatorBiasVOA 2.858.1.2.19
InterfacesTransceiverStatusValueFormatTxTurnUpStateDec 2.858.1.2.20
InterfacesTransceiverStatusValueFormatVendorSpecificFaultStatus 2.858.1.2.21

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.

2.858.1.2.4 InterfacesTransceiverStatusIntfValueFormatEgressAlarmStatus

alignmentMarkerLock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS alignment marker lock.
bipErrorsDetected	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS BIP errors detected.
blockLock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS block lock.
[changes]	*integer*	Changes since system boot.
erroredBlocksDetected	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS errored blocks detected.
highBer	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS high BER.
[lastChange]	*floating point number*	Last update time in UTC.
loa	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS loss of alignment.
los	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS loss of signal.
rxLocalFault	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	RX local fault.
rxRemoteFault	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	RX remote fault.

2.858.1.2.4.1 InterfacesTransceiverStatusValueFormatStr

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*string*	Current value.

2.858.1.2.5 InterfacesTransceiverStatusIntfValueFormatIngressAlarmStatus

alignmentMarkerLock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS alignment marker lock.
bipErrorsDetected	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS BIP errors detected.

blockLock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS block lock.
[changes]	*integer*	Changes since system boot.
erroredBlocksDetected	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS errored blocks detected.
[lastChange]	*floating point number*	Last update time in UTC.
loa	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS loss of alignment.
los	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	PCS loss of signal.
rxLocalFault	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	RX local fault.
rxRemoteFault	*InterfacesTransceiver-StatusValueFormatBool* see 2.858.1.2.1.1	RX remote fault.

2.858.1.2.6 InterfacesTransceiverStatusValueFormatFloat

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*floating point number*	Current value.

2.858.1.2.7 InterfacesTransceiverStatusValueFormatHex

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*integer*	Current value.

2.858.1.2.8 InterfacesTransceiverStatusValueFormatInt

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*integer*	Current value.

2.858.1.2.9 InterfacesTransceiverStatusValueFormatLaneAlignment

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
modemLockFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX alignment modem lock fault.
modemSyncDetectFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX alignment modem sync detect fault.

netLaneTxAlignment	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Network lane TX alignment.

2.858.1.2.10 InterfacesTransceiverStatusValueFormatModuleFaultStatus

[changes]	*integer*	Changes since system boot.
checkSum	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module fault status checksum.
[lastChange]	*floating point number*	Last update time in UTC.
modOverTempFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module fault status module over temperature fault.
modSpecFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module fault status module specific hw fault.
pldFlashInitFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module fault status pld flash init fault.
powerSupplyFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module fault status power supply fault.

2.858.1.2.11 InterfacesTransceiverStatusValueFormatModuleGeneralStatus

[changes]	*integer*	Changes since system boot.
hiPowerOn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status high power on.
hwInterlock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status hw interlock.
[lastChange]	*floating point number*	Last update time in UTC.
outOfAlgn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status out of alignment.
pmIntervalDone	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status PM interval done.
rxLOS	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status RX loss of signal.
rxNetworkLOL	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status RX network loss of lock.
txHostLOL	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status TX host loss of lock.
txLOF	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module general status TX loss of signal functionality.

2.858.1.2.12 InterfacesTransceiverStatusValueFormatModuleStateDec

[changes]	*integer*	Changes since system boot.
fault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state fault.
hiPwrDown	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state high power down.
hiPwrUp	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state high power up.
init	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state initialization.
[lastChange]	*floating point number*	Last update time in UTC.
lowPwr	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state low power.
ready	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state ready.
txOff	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state TX off.
txTurnOff	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state TX turn off.
txTurnOn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Module state TX turn on.

2.858.1.2.13 InterfacesTransceiverStatusValueFormatNetworkLaneFaultStatus

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
rxFIFOFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX FIFO fault.
rxLOL	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX LOL.
rxLOS	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX LOS.
rxTECFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX TEC fault.
tecFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TEC fault.

txLOL	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TX LOL.
txLOSF	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TX LOSF.
wavelengthUnlockFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Wavelength unlock fault.

2.858.1.2.14 InterfacesTransceiverStatusValueFormatNetworkLaneRxOTNStatus

[changes]	*integer*	Changes since system boot.
fastPMRxSignalDegradeAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Fast PM RX signal degrade alarm.
fastPMRxSignalFailAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Fast PM RX signal fail alarm.
fastPMTxSignalDegradeAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Fast PM TX signal degrade alarm.
fastPMTxSignalFailAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Fast PM TX signal fail alarm.
ingressFDDAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Ingress FDD alarm.
ingressFEDAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Ingress FED alarm.
[lastChange]	*floating point number*	Last update time in UTC.
lossOfFrame	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Loss of frame.
lossOfMultiframe	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Loss of multiframe.
oduAISMaintSignal	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	ODU AIS maintenance signal.
oduLCKMaintSignal	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	ODU LCK maintenance signal.
outOfFrame	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Out of frame.
outOfMultiframe	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Out of multiframe.
sectionMonitoringBackDefect	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Section monitoring back defect.

2.858.1.2.15 InterfacesTransceiverStatusValueFormatNetworkLaneVendorFAWS

adcOutputLow	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	ADC output low.
[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
modBiasConvergeFault	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Modulator bias converge fault.
oaPumpBiasHighAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	OA pump bias high alarm.
oaPumpBiasHighWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	OA pump bias high warn.
oaPumpBiasLowAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	OA pump bias low alarm.
oaPumpBiasLowWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	OA pump bias low warn.
rxPhaseCtrlLoopHighAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX phase control loop high alarm.
rxPhaseCtrlLoopHighWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX phase control loop high warn.
rxPhaseCtrlLoopLowAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX phase control loop low alarm.
rxPhaseCtrlLoopLowWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX phase control loop low warn.
rxTunedChannelPwrHighAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX tuned channel power high alarm.
rxTunedChannelPwrHighWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX tuned channel power high warn.
rxTunedChannelPwrLowAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX tuned channel power low alarm.
rxTunedChannelPwrLowWarn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	RX tuned channel power low warn.
txModBiasVOAHighAlarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TX mod bias VOA control loop high alarm.

txModBiasVOAHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX mod bias VOA control loop high warn.

2.858.1.2.16 InterfacesTransceiverStatusValueFormatRxTurnUpStateDec

adcOutput	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state ADC output.
[changes]	*integer*	Changes since system boot.
dispersionLock	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state dispersion lock.
[lastChange]	*floating point number*	Last update time in UTC.
rxDemodLock	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state demodulator lock.
rxInit	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state path initialization.
rxLoLaserReady	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state RX LO laser ready.
rxWaitForInput	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX turn up state wait for signal input.

2.858.1.2.17 InterfacesTransceiverStatusValueFormatScientificFloat

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*floating point number*	Current value.

2.858.1.2.18 InterfacesTransceiverStatusValueFormatTrib

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
state	*integer*	Current value.

2.858.1.2.19 InterfacesTransceiverStatusValueFormatTxModulatorBiasVOA

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
loop1Alarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 1 alarm.
loop1Warn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 1 warn.

loop2Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 2 alarm.
loop2Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 2 warn.
loop3Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 3 alarm.
loop3Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 3 warn.
loop4Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 4 alarm.
loop4Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 4 warn.
loop5Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 5 alarm.
loop5Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 5 warn.
loop6Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 6 alarm.
loop6Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 6 warn.
loop7Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 7 alarm.
loop7Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 7 warn.
loop8Alarm	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 8 alarm.
loop8Warn	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	Net mod bias VOA loop 8 warn.

2.858.1.2.20 InterfacesTransceiverStatusValueFormatTxTurnUpStateDec

[changes]	*integer*	Changes since system boot.
[lastChange]	*floating point number*	Last update time in UTC.
txDataPathLock	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state data path lock.
txInit	*InterfacesTransceiver-StatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state path initialization.

txLasReadyOff	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state TX laser ready off.
txLaserReady	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state TX laser ready.
txModulatorConverge	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state TX modulator converge.
txOutPwrAdj	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX turn up state TX output power adjustment.

2.858.1.2.21 InterfacesTransceiverStatusValueFormatVendorSpecificFaultStatus

adcCalibrationFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	ADC calibration fault.
apexRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	APEX RW fault.
bootEepromCRCFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Boot EEPROM CRC fault.
[changes]	*integer*	Changes since system boot.
cpldDevRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	CPLD device RW fault.
cryptoDevRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Crypto device RW fault.
ctrlProcImgABBootFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Controller proc image AB boot fault.
ctrlProcIntCommFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Controller proc internal comm fault.
dacCalibrationFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	DAC calibration fault.
dacRefClockRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	DAC reference clock RW fault.
devInitFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Device initialization fault.
fwHwMismatchFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	FW HW mismatch fault.
[lastChange]	*floating point number*	Last update time in UTC.
modemProcIntCommFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Modem proc internal comm fault.

opticsProcIntCommFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Optics proc internal comm fault.
picDriverRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	PIC driver RW fault.
picEepromCRCFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	PIC EEPROM CRC fault.
picEepromRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	PIC EEPROM RW fault.
picXTiaRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	PIC X TIA RW fault.
picYTiaRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	PIC Y TIA RW fault.
powerSupplyRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Power supply RW fault.
rxITLAFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX ITLA fault.
txITLAFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX ITLA fault.
viMonitorDevRWFault	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	VI monitor device RW fault.

2.858.1.3 InterfacesTransceiverStatusExtCfp2Dco

adcFaultCode	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	ADC fault code.
controllerState	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Controller state.
controllerSubState	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Controller substate.
dacFaultCode	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	DAC fault code.
laneAlignment	*InterfacesTransceiverStatusValueFormatLaneAlignment* see 2.858.1.2.9	TX/RX alignment.
laserFaultCode	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Laser fault code.

modulatorBiasFaultCode	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Modulator bias fault code.
modulatorConvergenceIter	*InterfacesTransceiverStatusValueFormatInt* see 2.858.1.2.8	Modulator convergence iteration count.
modulatorConvergenceStep	*InterfacesTransceiverStatusValueFormatInt* see 2.858.1.2.8	Modulator convergence step.
moduleFaultStatus	*InterfacesTransceiverStatusValueFormatModuleFaultStatus* see 2.858.1.2.10	Module fault status for cfp2Dco Dp04 module.
moduleFaults	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Module fault status.
moduleGeneral	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Module general status.
moduleGeneralStatus	*InterfacesTransceiverStatusValueFormatModuleGeneralStatus* see 2.858.1.2.11	Module general status for cfp2Dco Dp04 module.
moduleState	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Module state.
moduleStateDec	*InterfacesTransceiverStatusValueFormatModuleStateDec* see 2.858.1.2.12	Module state deciphered for cfp2Dco Dp04 module.
netLaneFaultStatus	*InterfacesTransceiverStatusValueFormatNetworkLaneFaultStatus* see 2.858.1.2.13	Network lane fault/status.
netLaneVendorSpecificFAWS	*InterfacesTransceiverStatusValueFormatNetworkLaneVendorFAWS* see 2.858.1.2.15	Network lane vendor specific fault/status.
networkLaneRxOTNStatus	*InterfacesTransceiverStatusValueFormatNetworkLaneRxOTNStatus* see 2.858.1.2.14	Network lane RX OTN status.
networkTxModulatorBiasVoaAWS	*InterfacesTransceiverStatusValueFormatTxModulatorBiasVOA* see 2.858.1.2.19	Network TX modulator bias VOA AWS.
networkVendorFaults	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Network vendor-specific faults/status.
rxLaserFaultCode	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	RX laser fault code.

rxLaserPowerHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser power high alarm.
rxLaserPowerHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser power high warn.
rxLaserPowerLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser power low alarm.
rxLaserPowerLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser power low warn.
rxLaserTempHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser temperature high alarm.
rxLaserTempHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser temperature high warn.
rxLaserTempLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser temperature low alarm.
rxLaserTempLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX laser temperature low warn.
rxPowerHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX total power high alarm.
rxPowerHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX total power high warn.
rxPowerLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX total power low alarm.
rxPowerLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	RX total power low warn.
rxTurnUpState	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	RX turn-up state.
rxTurnUpStateDec	*InterfacesTransceiverStatusValueFormatRxTurnUpStateDec* see 2.858.1.2.16	RX turn up state deciphered for cfp2Dco Dp04 module.
sopFastTracking	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	State of Polarization (SOP) fast tracking.
txLaserPowerHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser power high alarm.
txLaserPowerHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser power high warn.

txLaserPowerLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser power low alarm.
txLaserPowerLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser power low warn.
txLaserTempHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser temperature high alarm.
txLaserTempHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser temperature high warn.
txLaserTempLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser temperature low alarm.
txLaserTempLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX laser temperature low warn.
txModBiasHighAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX modulator bias high alarm.
txModBiasHighWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX modulator bias high warn.
txModBiasLowAlarm	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX modulator bias low alarm.
txModBiasLowWarn	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	TX modulator bias low warn.
txTurnUpState	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	TX turn-up state.
txTurnUpStateDec	*InterfacesTransceiverStatusValueFormatTxTurnUpStateDec* see 2.858.1.2.20	TX turn up state deciphered for cfp2Dco Dp04 module.
vendorFaults	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Vendor specific faults/status.
vendorSpecificFaultStatus	*InterfacesTransceiverStatusValueFormatVendorSpecificFaultStatus* see 2.858.1.2.21	Vendor specific fault status.

2.858.1.4 InterfacesTransceiverStatusHostLaneSpecific

[errFramesCurHost]	*InterfacesTransceiverStatusValueFormatScientificFloat* see 2.858.1.2.17	Current value of host-side post-FEC errored frames ratio.
[preFecBERCurrHost]	*InterfacesTransceiverStatusValueFormatScientificFloat* see 2.858.1.2.17	Current value of host-side pre-FEC bit error rate.
[rxOutputSignalStats]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Receive output signal status.
[txAdaptiveInputEqFault]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit adaptive input equalization fault.
[txCdrLol]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit clock and data recovery loss of lock.
[txLos]	*InterfacesTransceiverStatusValueFormat* see 2.858.1.2.3	Transmit loss of signal.

2.858.1.5 InterfacesTransceiverStatusInterfaceSpecific

[cfp2Dco]	*InterfacesTransceiverStatusExtCfp2DcoIntf* see 2.858.1.5.1	CFP2-DCO specific attributes.
[channels]	*table(numeric string) = InterfacesTransceiverStatusChannelSpecific* see 2.858.1.2	Channel specific attributes indexed by channel number.
[hostLanes]	*table(numeric string) = InterfacesTransceiverStatusHostLaneSpecific* see 2.858.1.4	Host lane specific attributes indexed by lane number.
[interface]	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Interface name.
[operSpeed]	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Operational speed of the interface.

2.858.1.5.1 InterfacesTransceiverStatusExtCfp2DcoIntf

egressAlarmStatus	*InterfacesTransceiverStatusIntfValueFormatEgressAlarmStatus* see 2.858.1.2.4	Client egress alarms/status.
fec	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Client forward error correction type.

fecUncorrectedBlocks	*InterfacesTransceiverStatusValueFormatInt* see 2.858.1.2.8	Client FEC uncorrected blocks.
ingressAlarmStatus	*InterfacesTransceiverStatusIntfValueFormatIngressAlarmStatus* see 2.858.1.2.5	Client ingress alarms/status.
laneCount	*InterfacesTransceiverStatusValueFormatInt* see 2.858.1.2.8	Client speed PMD lane count.
pcsAlarmStatus	*InterfacesTransceiverStatusValueFormatHex* see 2.858.1.2.7	Client physical coding sublayer alarm status.
preFecBer	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Client pre-FEC bit error rate.
rsFecCodewordSize	*InterfacesTransceiverStatusValueFormatInt* see 2.858.1.2.8	Client Reed-Solomon FEC codeword size.
speed	*InterfacesTransceiverStatusValueFormatStr* see 2.858.1.2.4.1	Client operational speed.
tributary	*InterfacesTransceiverStatusValueFormatTrib* see 2.858.1.2.18	Optical tributary.

2.859 show lldp [...]

enabledTlvTypes	*list of TlvType* see 2.859.2	Enabled optional TLVs.
lldpInterfaces	*table(Interface name) = LldpInterface* see 2.859.1	Lldp enabled Interfaces.
[managementAddressTransmitted]	*string*	Management Address used in the TLV.
managementAddressVrf	*string*	LLDP Management Address VRF.
messageTxHoldTime	*integer*	The holdtime in seconds inserted in the transmitted LLDP frames.
messageTxInterval	*integer*	The interval in seconds at which LLDP frames are transmitted.
reinitDelay	*integer*	The delay in seconds before LLDP reinitialization is attempted.

2.859.1 LldpInterface

rxEnabled	*boolean*	Indicates whether LLDP frames can be received on this interface.
txEnabled	*boolean*	Indicates whether LLDP frames can be sent from this interface.

2.859.2 TlvType

tlvType	*enum: portVlan* | *systemName* | *powerViaMdi* | *medCapabilities* | *linkAggregation8021* | *linkAggregation8023* | *managementAddress* | *portDescription* | *medNetworkPolicy* | *systemCapabilities* | *portVlanName* | *medLocationIdentification* | *systemDescription* | *maxFrameSize*	portDescription – Port Description TLV, systemName – System Name TLV, systemDescription – System Description TLV, systemCapabilities – System Capabilities TLV, managementAddress – Management Address TLV, portVlan – IEEE802.1 Port VLAN ID TLV, portVlanName – IEEE802.1 Port VLAN Name TLV, linkAggregation8021 – IEEE802.1 Link Aggregation TLV, linkAggregation8023 – IEEE802.3 Link Aggregation TLV, maxFrameSize – IEEE802.3 Maximum Frame Size TLV, powerViaMdi – IEEE802.3 Power Via MDI TLV, medCapabilities – LLDP-MED Capabilities TLV, medNetworkPolicy – LLDP-MED Network Policy TLV, medLocationIdentification – LLDP-MED Location Identification TLV,.

2.860 show lldp counters [...]

interfaces	*table(string) = LldpInterfaceTraffic* see 2.860.1	Lldp Interface Counters.

2.860.1 LldpInterfaceTraffic

rxDiscards	*integer*	The number of LLDP frames received and then discarded.
rxErrors	*integer*	The number of LLDP frames received with one or more detectable errors.
rxFrames	*integer*	The number of LLDP frames received.
tlvsDiscarded	*integer*	The number of LLDP TLVs discarded.
tlvsUnknown	*integer*	The number of LLDP TLVs received on this port that are not recognized.
txFrames	*integer*	The number of LLDP frames transmitted.
txFramesLengthExceeded	*integer*	The number of LLDP frames transmitted on this port that could not hold all the desired information.

2.861 show lldp local-info [...]

chassisId	*string*	Chassis identifier string.
chassisIdType	*enum: chassisComponent | networkAddress | interfaceName | local | unknown | interfaceAlias | portComponent | macAddress | reserved*	Type of the chassis.
localInterfaceInfo	*table(Interface name) = LocalInterfaceInfo* see 2.861.1	A mapping between local interface and local Interface information such as interface id, description, sent TLVs etc.
managementAddresses	*list of ManagementAddr* see 2.861.2	A list of management addresses ordered by priority.
[systemCapabilities]	*SystemCapabilities* see 2.861.3	Supported system capabilities and information about whether they are enabled.
[systemDescription]	*string*	System description.
[systemName]	*string*	System name.

2.861.1 LocalInterfaceInfo

[autoclassCompleted] **	*boolean*	PSE has concluded the Autoclass measurement. **Since: 4.23.0
[autoclassRequest] **	*boolean*	PD has requested an Autoclass measurement by the PSE. **Since: 4.23.0
[controlAbility] **	*boolean*	Indicates whether the pair selection can be controlled on neighbor interface. **Since: 4.21.3
[dualSignaturePowerClass-ExtModeA] **	*enum: powerClass1 | powerClass2 | powerClass4 | singleSignaturePd | twoPairOnlyPse | powerClass5 | powerClass3*	Power class on Mode A. **Since: 4.23.0
[dualSignaturePowerClass-ExtModeB] **	*enum: powerClass1 | powerClass2 | powerClass4 | singleSignaturePd | twoPairOnlyPse | powerClass5 | powerClass3*	Power class on Mode B. **Since: 4.23.0
[interfaceDescription]	*string*	Description of local/neighbor interface.
interfaceId	*string*	Name of local/neighbor interface(may contain escaped quotation marks).
interfaceIdType	*enum: agentCircuitId | networkAddress | interfaceName | local | unknown | interfaceAlias | portComponent | macAddress | reserved*	Type of local/neighbor interface.
interfaceId_v2 **	*string*	Name of local/neighbor interface. **Since: 4.24.2

[linkAggregation8021-InterfaceId] **	*integer*	Link Aggregation 802.1 interface identifier. **Since: 4.29.2
[linkAggregation8021Status] **	*enum: capableAndDisabled | capableAndEnabled | notCapable*	Link Aggregation 802.1 Status. **Since: 4.29.2
[linkAggregation8023-InterfaceId] **	*integer*	Link Aggregation 802.3 interface identifier. **Since: 4.29.2
[linkAggregation8023Status] **	*enum: capableAndDisabled | capableAndEnabled | notCapable*	Link Aggregation 802.3 Status. **Since: 4.29.2
[maxFrameSize]	*integer*	Indicates the maximum supported frame size in octets on the given interface.
[medInfo] **	*LldpMedInfo* see 2.861.1.1	LLDP-MED Info. **Since: 4.21.3
[pd4Pid] **	*boolean*	The PD supports powering of both Modes simultaneously. **Since: 4.23.0
[pdLoad] **	*boolean*	PD is dual-signature and power demand on Mode A and Mode B are electrically isolated. **Since: 4.23.0
[pdPoweredStatus] **	*enum: twoPairPowering-DualSignature | poweredSingleSignature | fourPairPoweringDual-Signature*	Existing powered configuration of the PD. **Since: 4.23.0
[pdRequestedPowerValue] **	*floating point number*	Power requested by device in watts. **Since: 4.21.3
[pdRequestedPowerValueMode-A] **	*floating point number*	Maximum input power (in watts) the PD is requesting for the A Mode. **Since: 4.23.0
[pdRequestedPowerValueMode-B] **	*floating point number*	Maximum input power (in watts) the PD is requesting for the B Mode. **Since: 4.23.0
[portClass] **	*enum: pClassPSE | pClassPD | unknown*	Port class of neighbor interface. **Since: 4.21.3
[portVlanId]	*integer*	Port VLAN ID.
[powerClass] **	*integer*	Power classification of neighbor interface. **Since: 4.21.3
[powerClassExt] **	*enum: powerClassExt7 | powerClassExt6 | powerClassExt8 | powerClassExt5 | powerClassExt1 | powerClassExt4 | powerClassExt3 | dualSignaturePd | powerClassExt2*	Power class ext. **Since: 4.23.0
[powerDownRequest] **	*boolean*	PD has requested a power down. **Since: 4.23.0

[powerDownTime] **	*integer*	Time in seconds the PD requests to be unpowered. Zero means to remain unpowered indefinitely. **Since: 4.23.0
[powerOverEthernetEnabled] **	*boolean*	Indicates whether MDI power is enabled on neighbor interface. **Since: 4.21.3
[powerOverEthernet-Supported] **	*boolean*	Indicates whether MDI power is supported on neighbor interface. **Since: 4.21.3
[powerPairs] **	*enum: signal* | *unknown* | *spare*	PSE power pairs on use on neighbor interface. **Since: 4.21.3
[powerPriority] **	*enum: high* | *low* | *critical* | *unknown*	Power priority on neighbor interface. **Since: 4.21.3
[powerSource] **	*enum: backup* | *unknown* | *pseAndLocal* | *primary* | *pse* | *reserved*	Power Source. **Since: 4.21.3
[powerType] **	*enum: type2PdDevice* | *type1PseDevice* | *type1PdDevice* | *type2PseDevice*	Power Type - type 1/2 and Power Sourcing Entity/Powered Device. **Since: 4.21.3
[powerTypeExt] **	*enum: type4DualSignaturePd* | *type4SingleSignaturePd* | *type3DualSignaturePd* | *type3Pse* | *type3SingleSignaturePd* | *type4Pse*	Power Type ext. **Since: 4.23.0
[pseAllocatedPowerValue] **	*floating point number*	Power granted to device in watts. **Since: 4.21.3
[pseAllocatedPowerValue-AlternativeA] **	*floating point number*	Maximum output power (in watts) the PSE is providing for the A Mode. **Since: 4.23.0
[pseAllocatedPowerValue-AlternativeB] **	*floating point number*	Maximum output power (in watts) the PSE is providing for the B Mode. **Since: 4.23.0
[pseAutoclassSupport] **	*boolean*	PSE supports Autoclass. **Since: 4.23.0
[pseMaximumAvailablePower] **	*floating point number*	Highest power (in watts) the PSE can grant. **Since: 4.23.0
[psePowerPairsExt] **	*enum: alternativeA* | *alternativeB* | *bothAlternatives*	Alternative used by PSE. **Since: 4.23.0
[psePoweringStatus] **	*enum: fourPairPowering-SingleSignature* | *twoPairPowering* | *poweringStatusReserved0* | *fourPairPoweringDual-Signature*	Existing powering configuration of the PSE. **Since: 4.23.0
unknownOrgDefinedTlvs	*list of UnknownOrgDefTlv* see 2.861.1.2	A list of unknown organizationally specific TLVs.

vlanNames **	*table(numeric string) = string*	A mapping between vlan id and vlan name. **Since: 4.26.0
vlanNamesChanged **	*boolean*	List of VLAN Name TLVs might not be up to date. **Since: 4.26.2
vlanNamesIncomplete **	*boolean*	List of VLAN Name TLVs is not complete. **Since: 4.26.2
[ztpBootVlan] **	*integer*	ZTP Boot VLAN. **Since: 4.26.0

2.861.1.1 LldpMedInfo

[assetIDTlvInfo] **	*string*	Asset ID Tlv. **Since: 4.22.0
[capabilities]	*MedCapabilitiesInfo* see 2.861.1.1.1	MED Capabilities.
[firmwareRevisionTlvInfo] **	*string*	Firmware Revision Tlv. **Since: 4.22.0
[hardwareRevisionTlvInfo] **	*string*	Hardware Revision Tlv. **Since: 4.22.0
[locationId] **	*MedLocationIdentification-Info* see 2.861.1.1.2	Location Identification information. **Since: 4.22.0
[manufacturerNameTlvInfo] **	*string*	Manufacturer Name Tlv. **Since: 4.22.0
[modelNameTlvInfo] **	*string*	Model Name Tlv. **Since: 4.22.0
networkPolicies	*list of MedNetworkPolicyInfo* see 2.861.1.1.3	Collection of Network Policies.
[serialNumberTlvInfo] **	*string*	Serial Number Tlv. **Since: 4.22.0
[softwareRevisionTlvInfo] **	*string*	Software Revision Tlv. **Since: 4.22.0

2.861.1.1.1 MedCapabilitiesInfo

capabilities	*boolean*	System supports 'Capabilities' TLV.
deviceType	*enum: endpointClass3 | networkConnectivity | endpointClass2 | endpointClass1 | notDefined | reserved*	MED device type.
extendedPd	*boolean*	System supports 'Extended Power via MDI - PD' TLV.
extendedPse	*boolean*	System supports 'Extended Power via MDI - PSE' TLV.
inventory	*boolean*	System supports 'Inventory' TLV.
location	*boolean*	System supports 'Location Identification' TLV.
networkPolicy	*boolean*	System supports 'Network Policy' TLV.

2.861.1.1.2 MedLocationIdentificationInfo

[civicAddress]	*CivicAddress* see 2.861.1.1.2.1	Civic Address LCI as per draft-ietf-geopriv-dhcp-civil-09.
[coordinateBased]	*CoordinateBasedLocationInfo* see 2.861.1.1.2.3	Coordinate-based LCI info as per RFC 3825.
[ecsElin]	*string*	Emergency Location Identification Number to support traditional PSAP-based Emergency Communications Systems.

2.861.1.1.2.1 CivicAddress

civicAddressElements	*list of CivicAddressElement* see 2.861.1.1.2.2	List of Civic Address Elements.
countryCode	*string*	Two-letter ISO 3166 country code.
what	*integer*	Which location the address refers to as per RFC 4776.

2.861.1.1.2.2 CivicAddressElement

caType	*integer*	Civic Address (CA) Type as per draft-ietf-geopriv-dhcp-civil-09.
caValue	*string*	Civic Address (CA) Value as per draft-ietf-geopriv-dhcp-civil-09.

2.861.1.1.2.3 CoordinateBasedLocationInfo

aT	*integer*	Altitude type (at) as per RFC 3825.
altRes	*integer*	Altitude resolution as per RFC 3825.
altitude	*integer*	Altitude as per RFC 3825: fixed point value with 8 bits fractional part.
datum	*integer*	Datum as per RFC 3825.
laRes	*integer*	Latitude resolution as per RFC 3825.
latitude	*integer*	Latitude as per RFC 3825: fixed point value with 25 bits fractional part.
loRes	*integer*	Longitude resolution as per RFC 3825.
longitude	*integer*	Longitude as per RFC 3825: fixed point value with 25 bits fractional part.

2.861.1.1.3 MedNetworkPolicyInfo

applicationType	*integer*	Application Type for given policy.
dscpValue	*integer*	DSCP value to be used by Endpoint Device for given application to specify priority for IP datagram.
[layer2Priority]	*integer*	Layer 2 priority to be used for the specified application type.
tagged	*boolean*	The specified application type is using a 'tagged' VLAN.

unknownPolicy	*boolean*	Endpoint Device wants to explicitly advertise that this policy is required by the device, but is currently unknown.
[vlanId]	*integer*	VLAN identifier for the port as defined in IEEE 802.1Q-2003.

2.861.1.2 UnknownOrgDefTlv

[malformedTlvError]	*string*	Error string if the TLV is malformed.
[ouiStr]	*string*	Hexdump of Organizationally Unique Identifier of the information received from the neighbor.
[subType]	*integer*	Unknown Organizationally-Defined TLV type.
tlvLines	*list of string*	Represents the received Unknown Organizationally-Defined TLV in either decoded string format or hexdump depending on whether decoding was successful.

2.861.2 ManagementAddr

address	*string*	LLDP management address which could either be an IPv4/v6 address or a MAC address, may be used to reach higher layer entities.
addressType	*enum: ipv4* | *ipv6* | *unknown* | *ethernet*	Type of LLDP management address.
interfaceNum	*integer*	Interface number associated with the management address, in the encoding specified by interfaceNumType attribute.
interfaceNumType	*enum: badType* | *ifIndex* | *systemPortNumber* | *unknown*	Interface numbering method used by the interfaceNum attribute.
[interfaceNumTypeValue]	*integer*	Interface number subtype value which is unrecognized.
oidString	*string*	OID value used to identify the type of hardware component or protocol entity associated with the management address.

2.861.3 SystemCapabilities

[bridge]	*boolean*	Indicates whether the system is enabled to be a bridge.
[docsisCableDevice]	*boolean*	Indicates whether the system is enabled to be a DOCSIS Cable Device.
[other]	*boolean*	Indicates whether the system has any capabilities other than repeater,bridge,wlanAccessPoint,router,telephone,docsisCableDev
[repeater]	*boolean*	Indicates whether the system is enabled to be a repeater.
[router]	*boolean*	Indicates whether the system is enabled to be a router.

[stationOnly]	*boolean*	Indicates whether the system is enabled to be a station and nothing else.
[telephone]	*boolean*	Indicates whether the system is enabled to be a telephone.
[wlanAccessPoint]	*boolean*	Indicates whether the system is enabled to be a WLAN Access Point.

2.862 show lldp neighbors [...]

lldpNeighbors	*list of LldpNeighbor* see 2.862.1	Lldp Neighbors.
tablesAgeOuts	*integer*	The number of times the information advertised by a MSAP has been deleted from tables contained in the LLDP remote systems MIB because the information timeliness interval has expired.
tablesDeletes	*integer*	The number of times the information advertised by a MSAP has been deleted from tables contained in the LLDP remote systems MIB.
tablesDrops	*integer*	The number of times the information advertised by a MSAP could not be entered into tables contained in the LLDP remote systems MIB.
tablesInserts	*integer*	The number of times the information advertised by a MSAP has been inserted into tables contained in the LLDP remote systems MIB.
[tablesLastChangeTime]	*floating point number*	The last time tables contained in the LLDP remote systems MIB are updated.

2.862.1 LldpNeighbor

neighborDevice	*string*	Name of the neighbor device.
neighborPort	*string*	Name of the neighbor port.
port	*interface name*	Name of the interface.
ttl	*integer*	The Time To Live value (in seconds) advertised by the remote system.

2.863 show lldp neighbors detail [...]

lldpNeighbors	*table(Interface name) = LldpNeighborsList* see 2.863.1	A mapping between local interface and neighbors discovered on that interface.

2.863.1 LldpNeighborsList

lldpNeighborInfo	*list of LldpNeighborInfo* see 2.863.1.1	List of LLDP neighbors' information.

2.863.1.1 LldpNeighborInfo

chassisId	*string*	Chassis identifier string.
chassisIdType	*enum: chassisComponent | networkAddress | interfaceName | local | unknown | interfaceAlias | portComponent | macAddress | reserved*	Type of the chassis.
lastChangeTime	*floating point number*	The UTC time at which the neighbor entry was last updated.
lastContactTime	*floating point number*	The UTC time at which the neighbor last communicated with the system.
managementAddresses	*list of ManagementAddr* see 2.861.2	List of management addresses ordered by priority.
neighborDiscoveryTime	*floating point number*	The UTC time at which the neighbor was created.
neighborInterfaceInfo	*NeighborInterfaceInfo* see 2.863.1.1.1	Information of neighbor interface like interface id, description along with all the TLVs received on it.
[systemCapabilities]	*SystemCapabilities* see 2.861.3	Supported system capabilities and information about whether they are enabled.
[systemDescription]	*string*	System description.
[systemName]	*string*	System name.
ttl	*integer*	The Time To Live value (in seconds) advertised by the neighbor.

2.863.1.1.1 NeighborInterfaceInfo

autoNegAdvertised-Capabilities	*list of string*	List of auto-negotiation capabilities advertised on neighbor interface.
[autoNegCapability]	*enum: capableAndDisabled | capableAndEnabled | notCapable*	Auto-negotiation capability of neighbor interface.
[autoclassCompleted] **	*boolean*	PSE has concluded the Autoclass measurement. **Since: 4.23.0
[autoclassRequest] **	*boolean*	PD has requested an Autoclass measurement by the PSE. **Since: 4.23.0
[controlAbility]	*boolean*	Indicates whether the pair selection can be controlled on neighbor interface.

[dualSignaturePowerClass-ExtModeA] **	*enum: powerClass1 | powerClass2 | powerClass4 | singleSignaturePd | twoPairOnlyPse | powerClass5 | powerClass3*	Power class on Mode A. **Since: 4.23.0
[dualSignaturePowerClass-ExtModeB] **	*enum: powerClass1 | powerClass2 | powerClass4 | singleSignaturePd | twoPairOnlyPse | powerClass5 | powerClass3*	Power class on Mode B. **Since: 4.23.0
[interfaceDescription]	*string*	Description of local/neighbor interface.
interfaceId	*string*	Name of local/neighbor interface(may contain escaped quotation marks).
interfaceIdType	*enum: agentCircuitId | networkAddress | interfaceName | local | unknown | interfaceAlias | portComponent | macAddress | reserved*	Type of local/neighbor interface.
interfaceId_v2 **	*string*	Name of local/neighbor interface. **Since: 4.24.2
[linkAggregation8021-InterfaceId] **	*integer*	Link Aggregation 802.1 interface identifier. **Since: 4.29.2
[linkAggregation8021Status] **	*enum: capableAndDisabled | capableAndEnabled | notCapable*	Link Aggregation 802.1 Status. **Since: 4.29.2
[linkAggregation8023-InterfaceId] **	*integer*	Link Aggregation 802.3 interface identifier. **Since: 4.29.2
[linkAggregation8023Status] **	*enum: capableAndDisabled | capableAndEnabled | notCapable*	Link Aggregation 802.3 Status. **Since: 4.29.2
[managementVid]	*integer*	Management vlan id.
[maxFrameSize]	*integer*	Indicates the maximum supported frame size in octets on the given interface.
[medInfo] **	*LldpMedInfo* see 2.861.1.1	LLDP-MED Info. **Since: 4.21.3
[operMauType]	*string*	Operational Medium Attachment Unit type of the neighbor interface.
[pd4Pid] **	*boolean*	The PD supports powering of both Modes simultaneously. **Since: 4.23.0
[pdLoad] **	*boolean*	PD is dual-signature and power demand on Mode A and Mode B are electrically isolated. **Since: 4.23.0
[pdPoweredStatus] **	*enum: twoPairPowering-DualSignature | poweredSingleSignature | fourPairPoweringDual-Signature*	Existing powered configuration of the PD. **Since: 4.23.0

[pdRequestedPowerValue] **	*floating point number*	Power requested by device in watts. **Since: 4.21.3
[pdRequestedPowerValueMode-A] **	*floating point number*	Maximum input power (in watts) the PD is requesting for the A Mode. **Since: 4.23.0
[pdRequestedPowerValueMode-B] **	*floating point number*	Maximum input power (in watts) the PD is requesting for the B Mode. **Since: 4.23.0
portAndProtocolVlanEnabled	*table(numeric string) = boolean*	A mapping between Port And Protocol VLAN ID and whether it is enabled.
portAndProtocolVlan-Supported	*table(numeric string) = boolean*	A mapping between Port And Protocol VLAN ID and whether it is supported.
[portClass]	*enum: pClassPSE | pClassPD | unknown*	Port class of neighbor interface.
[portVlanId]	*integer*	Port VLAN ID.
[powerClass]	*integer*	Power classification of neighbor interface.
[powerClassExt] **	*enum: powerClassExt7 | powerClassExt6 | powerClassExt8 | powerClassExt5 | powerClassExt1 | powerClassExt4 | powerClassExt3 | dualSignaturePd | powerClassExt2*	Power class ext. **Since: 4.23.0
[powerDownRequest] **	*boolean*	PD has requested a power down. **Since: 4.23.0
[powerDownTime] **	*integer*	Time in seconds the PD requests to be unpowered. Zero means to remain unpowered indefinitely. **Since: 4.23.0
[powerOverEthernetEnabled]	*boolean*	Indicates whether MDI power is enabled on neighbor interface.
[powerOverEthernet-Supported]	*boolean*	Indicates whether MDI power is supported on neighbor interface.
[powerPairs]	*enum: signal | unknown | spare*	PSE power pairs on use on neighbor interface.
[powerPriority] **	*enum: high | low | critical | unknown*	Power priority on neighbor interface. **Since: 4.21.3
[powerSource] **	*enum: backup | unknown | pseAndLocal | primary | pse | reserved*	Power Source. **Since: 4.21.3
[powerType] **	*enum: type2PdDevice | type1PseDevice | type1PdDevice | type2PseDevice*	Power Type - type 1/2 and Power Sourcing Entity/Powered Device. **Since: 4.21.3
[powerTypeExt] **	*enum: type4DualSignaturePd | type4SingleSignaturePd | type3DualSignaturePd | type3Pse | type3SingleSignaturePd | type4Pse*	Power Type ext. **Since: 4.23.0

protocolIdentityInfo	*list of string*	Protocols enabled on neighbor interface.
[pseAllocatedPowerValue] **	*floating point number*	Power granted to device in watts. **Since: 4.21.3
[pseAllocatedPowerValue-AlternativeA] **	*floating point number*	Maximum output power (in watts) the PSE is providing for the A Mode. **Since: 4.23.0
[pseAllocatedPowerValue-AlternativeB] **	*floating point number*	Maximum output power (in watts) the PSE is providing for the B Mode. **Since: 4.23.0
[pseAutoclassSupport] **	*boolean*	PSE supports Autoclass. **Since: 4.23.0
[pseMaximumAvailablePower] **	*floating point number*	Highest power (in watts) the PSE can grant. **Since: 4.23.0
[psePowerPairsExt] **	*enum: alternativeA* | *alternativeB* | *bothAlternatives*	Alternative used by PSE. **Since: 4.23.0
[psePoweringStatus] **	*enum: fourPairPowering-SingleSignature* | *twoPairPowering* | *poweringStatusReserved0* | *fourPairPoweringDual-Signature*	Existing powering configuration of the PSE. **Since: 4.23.0
unknownOrgDefinedTlvs	*list of UnknownOrgDefTlv* see 2.861.1.2	A list of unknown organizationally specific TLVs.
unknownTlvs	*list of UnknownTlv* see 2.863.1.1.1.1	List of Unknown Tlvs which are outside the standard TLVs defined by LLDP protocol.
[vidUsageDigest]	*integer*	VID usage Digest associated with the neighbor.
vlanNames	*table(numeric string) = string*	A mapping between vlan id and vlan name.
[ztpBootVlan] **	*integer*	ZTP Boot VLAN. **Since: 4.26.0

2.863.1.1.1.1 UnknownTlv

[tlvLines]	*list of string*	Hexdump of the information aboutreceived Unknown TLV.
[tlvType]	*integer*	Unknown TLV type which is not defined in LLDP protocol.

2.864 show lldp traffic [...]

see 2.860

2.865 show load-balance destination port-channel PORT_CHANNEL { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-mac SRC_MAC) | (dst-mac DST_MAC) | (eth-type ETH_TYPE)) } | { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-ip-address SRC_IP) | (dst-ip-address DST_IP) | (ip-ttl IP_TTL) | (ip-protocol IP_PROTO)) | ((src-l4-port SRC_L4_PORT) | (dst-l4-port DST_L4_PORT)) } | { ((ingress-interface ING_INTF) | (vlan VLAN) | non-unicast) | ((src-ipv6-address SRC_IPV6) | (dst-ipv6-address DST_IPV6) | (hop-limit HOP_LIMIT) | (next-header NEXT_HDR) | (flow-label FLOW_LABEL)) | ((src-l4-port SRC_L4_PORT) | (dst-l4-port DST_L4_PORT)) }

`outputIntf`	*interface name*	Destination interface of the Port-Channel.

2.866 show load-balance ecmp sand fields [...]

revision 2

modules	*list of Module* see 2.866.1	Collection of linecard's ECMP hashing settings.

2.866.1 Module

ecmpHashIngressIntf	*boolean*	ECMP hashing on ingress interface.
ecmpHashKeyShift	*integer*	ECMP hash key shift.
ecmpHashPolynomial	*integer*	ECMP hash polynomial.
ecmpHashSeed	*integer*	ECMP hash seed.
featureErrorCodes	*table(string) = string*	Mapping of load-balance features to error codes.
gtp	*LoadBalanceFieldsGtp* see 2.866.1.1	GTP hash fields settings.
ipv4	*LoadBalanceFieldsIpv4* see 2.866.1.2	IPv4 hash fields settings.
ipv6	*LoadBalanceFieldsIpv6* see 2.866.1.3	IPv6 hash fields settings.
l4	*LoadBalanceFieldsL4* see 2.866.1.4	L4 hash fields settings.
mac	*LoadBalanceFieldsMac* see 2.866.1.5	MAC hash fields settings.
mpls	*LoadBalanceFieldsMpls* see 2.866.1.6	MPLS hash fields settings.
name	*string*	Name of the load balance profile or module.
pktTypeGueHashMode	*enum: inner-ip | inner-and-outer-ip | outer-ip*	GUE packet hash settings.
pktTypeIpOGueTransitOuterUdp **	*boolean*	IP over GUE transit outer UDP hash enabled. **Since: 4.28.0
pktTypeMplsOGre	*enum: inner-ip | inner-and-outer-ip | outer-ip*	MPLS over GRE packet hash settings.
symmetricHash	*boolean*	Full symmetric hash setting. When showing a linecard's load-balance policy, this becomes an estimate made by comparing the linecard's configuration with the expected full symmetric hashing settings.
udp **	*LoadBalanceFieldsUdp* see 2.866.1.7	UDP hash fields settings. **Since: 4.30.0

2.866.1.1 LoadBalanceFieldsGtp

teid	*boolean*	Hashing on TEID.

2.866.1.2 LoadBalanceFieldsIpv4

inner	*LoadBalanceFieldsIpv4-Header* see 2.866.1.2.1	Hashing on inner IPv4 header.
outer	*LoadBalanceFieldsIpv4-Header* see 2.866.1.2.1	Hashing on outer IPv4 header.

2.866.1.2.1 LoadBalanceFieldsIpv4Header

dst	*boolean*	Hashing on destination IPv4 address.
protocol	*boolean*	Hashing on Protocol.
src	*boolean*	Hashing on source IPv4 address.
[symmetric]	*boolean*	Symmetrize source and destination fields.
ttl	*boolean*	Hashing on Time-To-Live.

2.866.1.3 LoadBalanceFieldsIpv6

inner	*LoadBalanceFieldsIpv6-Header* see 2.866.1.3.1	Hashing on inner IPv6 header.
outer	*LoadBalanceFieldsIpv6-Header* see 2.866.1.3.1	Hashing on outer IPv6 header.

2.866.1.3.1 LoadBalanceFieldsIpv6Header

dst	*boolean*	Hashing on destination IPv6 address.
flowLabel	*boolean*	Hashing on Flow Label.
hopLimit	*boolean*	Hashing on Hop-Limit.
nextHeader	*boolean*	Hashing on Next Header.
src	*boolean*	Hashing on source IPv6 address.
[symmetric]	*boolean*	Symmetrize source and destination fields.

2.866.1.4 LoadBalanceFieldsL4

portDst	*boolean*	Hashing on destination port.
portSrc	*boolean*	Hashing on source port.
[symmetric]	*boolean*	Symmetrize source and destination fields.

2.866.1.5 LoadBalanceFieldsMac

inner	*LoadBalanceFieldsMac-Header* see 2.866.1.5.1	Hashing on inner MAC header.
outer	*LoadBalanceFieldsMac-Header* see 2.866.1.5.1	Hashing on outer MAC header.

2.866.1.5.1 LoadBalanceFieldsMacHeader

dst	*boolean*	Hashing on destination MAC address.
ethType	*boolean*	Hashing on ethertype.
src	*boolean*	Hashing on source MAC address.
[symmetric]	*boolean*	Symmetrize source and destination fields.
vlan	*boolean*	Hashing on vlan.

2.866.1.6 LoadBalanceFieldsMpls

entropyLabelEnabled	*boolean*	Hashing on Entropy label.
label	*boolean*	Hashing on MPLS label.

2.866.1.7 LoadBalanceFieldsUdp

dstPortPayload	*table(numeric string) = LoadBalanceUdpPayload* see 2.866.1.7.1	Payload hashing per destination port.

2.866.1.7.1 LoadBalanceUdpPayload

payloadBytes	*list of integer*	Payload bytes to hash.

2.867 show load-balance port-channel sand fields [...]

revision 3

modules	*list of Module* see 2.867.1	Collection of load-balance hashing profiles or linecard LAG hashing settings.

2.867.1 Module

Could also be an instance of:
AradProfile 2.867.2

egressReplicationLagLb	*boolean*	Load Balancing using egress replication.
featureErrorCodes	*table(string) = string*	Mapping of load-balance features to error codes.
gtp	*LoadBalanceFieldsGtp* see 2.866.1.1	GTP hash fields settings.
ipPktIpTcpUdp	*boolean*	L4 and IP header hash on LAG for IP packets.
ipPktMac	*boolean*	MAC header hash on LAG for IP packets.
ipv4	*LoadBalanceFieldsIpv4* see 2.866.1.2	IPv4 hash fields settings.
ipv6	*LoadBalanceFieldsIpv6* see 2.866.1.3	IPv6 hash fields settings.
l4	*LoadBalanceFieldsL4* see 2.866.1.4	L4 hash fields settings.
lagHashIngressIntf	*boolean*	LAG hashing on ingress interface.
lagHashKeyShift	*integer*	LAG hash key shift.
lagHashMode	*enum: flow-based | round-robin | clock-based*	packet distribution policy for LAGs.
lagHashPolynomial	*integer*	LAG hash polynomial.
lagHashSeed	*integer*	LAG hash seed.
lagPreferLocalMembers	*boolean*	Local LAG members programmed on each chip.
loadSharingPreferLocal-Supported	*boolean*	System supports prefer local setting.
mac	*LoadBalanceFieldsMac* see 2.866.1.5	MAC hash fields settings.
mpls	*LoadBalanceFieldsMpls* see 2.866.1.6	MPLS hash fields settings.
name	*string*	Name of the load balance profile or module.
pktTypeGueHashMode	*enum: inner-ip | inner-and-outer-ip | outer-ip*	GUE packet hash settings.
pktTypeIpOGueTransitOuter-Udp **	*boolean*	IP over GUE transit outer UDP hash enabled. **Since: 4.28.0
pktTypeMplsOGre	*enum: inner-ip | inner-and-outer-ip | outer-ip*	MPLS over GRE packet hash settings.
selectionMethods	*SelectionMethods* see 2.867.1.1	Methods to select member from LAG hash key.

symmetricHash	*boolean*	Full symmetric hash setting. When showing a linecard's load-balance policy, this becomes an estimate made by comparing the linecard's configuration with the expected full symmetric hashing settings.
udp **	*LoadBalanceFieldsUdp* see 2.866.1.7	UDP hash fields settings. **Since: 4.30.0
unadjustSwitchportFirst-Header	*boolean*	Do not adjust the starting header for calculating LAG hashing key on switchport.

2.867.1.1 SelectionMethods

multiplication	*boolean*	Multiply hash key by member number then shift.
smoothDivision	*boolean*	Use 8 LSB of hash key to look up table.

2.867.2 AradProfile

ecmpHashIngressIntf	*boolean*	ECMP hashing on ingress interface.
ecmpHashKeyShift	*integer*	ECMP hash key shift.
ecmpHashPolynomial	*integer*	ECMP hash polynomial.
ecmpHashSeed	*integer*	ECMP hash seed.
egressReplicationLagLb	*boolean*	Load Balancing using egress replication.
featureErrorCodes	*table(string) = string*	Mapping of load-balance features to error codes.
gtp	*LoadBalanceFieldsGtp* see 2.866.1.1	GTP hash fields settings.
ipPktIpTcpUdp	*boolean*	L4 and IP header hash on LAG for IP packets.
ipPktMac	*boolean*	MAC header hash on LAG for IP packets.
ipv4	*LoadBalanceFieldsIpv4* see 2.866.1.2	IPv4 hash fields settings.
ipv6	*LoadBalanceFieldsIpv6* see 2.866.1.3	IPv6 hash fields settings.
l4	*LoadBalanceFieldsL4* see 2.866.1.4	L4 hash fields settings.
lagHashIngressIntf	*boolean*	LAG hashing on ingress interface.
lagHashKeyShift	*integer*	LAG hash key shift.
lagHashMode	*enum: flow-based | round-robin | clock-based*	packet distribution policy for LAGs.
lagHashPolynomial	*integer*	LAG hash polynomial.
lagHashSeed	*integer*	LAG hash seed.
lagPreferLocalMembers	*boolean*	Local LAG members programmed on each chip.
loadSharingPreferLocal-Supported	*boolean*	System supports prefer local setting.
mac	*LoadBalanceFieldsMac* see 2.866.1.5	MAC hash fields settings.
mpls	*LoadBalanceFieldsMpls* see 2.866.1.6	MPLS hash fields settings.
name	*string*	Name of the load balance profile or module.

pktTypeGueHashMode	*enum: inner-ip | inner-and-outer-ip | outer-ip*	GUE packet hash settings.
pktTypeIpOGueTransitOuter-Udp **	*boolean*	IP over GUE transit outer UDP hash enabled. **Since: 4.28.0
pktTypeMplsOGre	*enum: inner-ip | inner-and-outer-ip | outer-ip*	MPLS over GRE packet hash settings.
selectionMethods	*SelectionMethods* see 2.867.1.1	Methods to select member from LAG hash key.
sliceIds	*list of string*	Linecards that are associated with this profile.
symmetricHash	*boolean*	Full symmetric hash setting. When showing a linecard's load-balance policy, this becomes an estimate made by comparing the linecard's configuration with the expected full symmetric hashing settings.
udp **	*LoadBalanceFieldsUdp* see 2.866.1.7	UDP hash fields settings. **Since: 4.30.0
unadjustSwitchportFirst-Header	*boolean*	Do not adjust the starting header for calculating LAG hashing key on switchport.

2.868 show load-balance profile [...]

revision 3

see 2.867

2.869 show load-balance server hardware connection history summary

expiringEntries	*integer*	Total number of connection history entries expiring in the next minute.
nhGroupsUsedEntries	*table(string) = integer*	Mapping of nexthop group name to number of used connection history entries.
totalEntries	*integer*	Total number of connection history hardware entries.
usedEntries	*integer*	Total number of used connection history hardware entries.

2.870 show load-balance server hardware errors

policyErrors	*table(string) = LbPolicyErrors* see 2.870.1	A mapping of load-balance policy name to the errors in that policy.

2.870.1 LbPolicyErrors

matchErrors	*table(string) = LbMatchErrors* see 2.870.1.1	A mapping of load-balance match name to the errors in that match-rule.

2.870.1.1 LbMatchErrors

fragment	*boolean*	Flag specifying whether the match-rule is expected to match fragments.
port	*integer*	Destination port configured in the match-rule.
protocol	*string*	Protocol configured in the match-rule.
reason	*list of string*	List of reasons explaining the failure.
status	*string*	Hardware status of the match-rule.
virtualIp	*IP address prefix*	IP address configured in the match-rule.

2.871 show load-balance server match-rule

matchRules	*list of LoadBalancerMatchRule* see 2.871.1	List of match-rules applied across all the load-balance policies.

2.871.1 LoadBalancerMatchRule

[action]	*string*	Action name.
[actionType]	*string*	Type of action configured in the match-rule.
[fragment] **	*boolean*	Flag specifying whether the match-rule is expected to match fragments. **Since: 4.23.2
[matchName]	*string*	Name of the match-rule.
[policyName]	*string*	Name of the load-balance policy.
[port]	*integer*	Destination port configured in the match-rule.
[protocol]	*string*	Protocol configured in the match-rule.
[status]	*enum: outOfResources | notHandled | inProgress | inHardware | partiallyInHardware | duplicateRule*	Hardware status of the match-rule.
[virtualIp]	*IP address prefix*	IP address configured in the match-rule.
[vrfList]	*list of string*	List of VRFs that have this match-rule applied.

2.872 show load-balance server nexthop-group counters

lbCounterEntries	*table(string) = LbCounterEntry* see 2.872.1	Mapping of nexthop-group name to load-balanacer counters.

2.872.1 LbCounterEntry

daisyChainPktCount	*integer*	Total number of packets forwarded using connection-history.
fwdPktCount	*integer*	Total number of packets forwarded without connection-history.
tcpResetPktCount	*integer*	Total number of TCP connection resets.

2.873 show load-balance server nexthop-group counters rates

lbCounterRateEntries	*table(string) = LbCounterRateEntry* see 2.873.1	Mapping of nexthop-group name to load-balanacer counter rate.

2.873.1 LbCounterRateEntry

daisyChainPktRate	*floating point number*	Rate of packets forwarded using connection-history.
fwdPktRate	*floating point number*	Rate of packets forwarded without connection-history.
tcpResetPktRate	*floating point number*	Rate of TCP connection resets.

2.874 show load-balance server policy <policyName> match-rule <match-Name> [...]

see 2.871.1

2.875 show load-balance server policy [...]

vrfInfo	*table(string) = LoadBalancerVrfPolicy* see 2.875.1	A mapping of VRF name to the load-balance policy information in that VRF.

2.875.1 LoadBalancerVrfPolicy

policies	*table(string) = LoadBalancerPolicy* see 2.875.1.1	A mapping of VRF name to the load-balance policies applied in that VRF.

2.875.1.1 LoadBalancerPolicy

matchRules	*table(string) = LoadBalancerMatchRule* see 2.871.1	A mapping of load-balance policy name to the match-rules applied in that policy.

2.876 show load-balance server summary

duplicateMatchRules	*integer*	Number of duplicate match-rules.
matchRulesConfigured	*integer*	Total number of match-rules configured.
matchRulesPending	*integer*	Number of match-rules pending hardware programming.
matchRulesProgrammed	*integer*	Number of match-rules programmed in hardware.
policiesConfigured	*integer*	Total number of policies configured.
unprogrammedMatchRules	*integer*	Number of unprogrammed match-rules.
vrfSummary	*table(string) = LoadBalancerVrfSummary* see 2.876.1	Per VRF summary of load-balancer policies and match-rules.

2.876.1 LoadBalancerVrfSummary

matchRulesConfigured	*integer*	Total number of match-rules configured in this VRF.
matchRulesProgrammed	*integer*	Number of match-rules programmed in hardware on this VRF.
policiesApplied	*list of string*	List of policies applied in this VRF.
state	*enum: notPresent | down | active*	State of the VRF.

2.877 show loop-protection [...]

[detail]	*LoopProtectShowDetail* see 2.877.1	Loop protection detailed.
disableTime	*integer*	How long interface will be disabled, if a loop is detected, in seconds.
enabled	*boolean*	Loop Protection enabled.
rateLimit	*integer*	Throttling level of loop protection messages per second.
totalVlansEnabled	*integer*	Total number of vlans loop protection is enabled on.
transmitInterval	*integer*	Packet transmission interval per second.

2.877.1 LoopProtectShowDetail

destinationAddress	*MAC address*	Destination address for loop protection packets.
ethernetType	*0x88b7*	Ethernet type used by protocol.
receiveAction	*Interface Disable*	Trigger when a loop protection packet is received.
vlans	*table(numeric string) = LoopProtectShowDetail-Vlan* see 2.877.1.1	A dict with the mapping vlan to loop protect vlan information.

2.877.1.1 LoopProtectShowDetailVlan

disabledIntfs	*list of Interface name*	Interfaces disabled by loop protection.
[latestDisabledTime]	*floating point number*	Last time an interface was disabled on this vlan by loop protection.
[loopDetected]	*boolean*	Loop detected on Vlan.
totalIntfs	*integer*	Total number of interfaces in vlan protected by loop protection.

2.878 show loop-protection counters

enabled	*boolean*	Loop protection enabled.
intfCounters	*table(string) = LoopProtectCounter* see 2.878.1	List of counters per interface.
vlanCounters	*table(numeric string) = LoopProtectCounter* see 2.878.1	List of counters per vlan.

2.878.1 LoopProtectCounter

rx	*integer*	Packets received.
rxOther	*integer*	Packets received in error.
tx	*integer*	Packets transmitted.

2.879 show loop-protection vlan VLANSET

enabled	*boolean*	Loop protection enabled.
vlanIntfs	*table(numeric string) = LoopProtectVlanIntfList* see 2.879.1	Loop protect vlan information.

2.879.1 LoopProtectVlanIntfList

intfs	*list of LoopProtectIntfInfo* see 2.879.1.1	Loop protect information for interfaces in this vlan.

2.879.1.1 LoopProtectIntfInfo

[disabledAt]	*floating point number*	When the interface was disabled.
[disabledUntil]	*floating point number*	When loop protection is scheduled to bring this interface up.
[intf]	*interface name*	Loop protection information about this interface.
[lpDisabled]	*boolean*	Loop detected on Vlan.
lpEnabled	*boolean*	Whether this interface is protected on loop protection.
[state]	*enum: shutdown | unknownEnabledState | enabled*	Current state of the interface.

2.880 show mac access-lists [...]

see 2.241.1.2.1.1

2.881 show mac pause-frame

`pauseFramePassThrough`	*boolean*	Pause frame visibility.

2.882 show mac security counters [...]

interfaces	*table(Interface name)* = *MacSecCountersInterface* see 2.882.1	A mapping between an interface and MAC security counters.

2.882.1 MacSecCountersInterface

[countersDetail]	*MacSecIntfCountersDetail* see 2.882.1.1	MAC security interface counter detailed information.
inOctetsDecrypted	*integer*	Number of received octets decrypted.
inPktsDecrypted	*integer*	Number of received packets decrypted.
outOctetsEncrypted	*integer*	Number of transmitted octets encrypted.
outPktsEncrypted	*integer*	Number of transmitted packets encrypted.

2.882.1.1 MacSecIntfCountersDetail

[inBroadcastPkts] **	*integer*	Number of received broadcast packets on the controlled port. **Since: 4.28.2
[inMulticastPkts] **	*integer*	Number of received multicast packets on the controlled port. **Since: 4.28.2
[inOctets] **	*integer*	Number of received octets on the controlled port. **Since: 4.28.2
[inOctetsProtected] **	*integer*	Number of received octets that were ICV protected but unencrypted. **Since: 4.29.1
inPktsBadTag	*integer*	Number of received packets with bad security tag.
inPktsCtrl	*integer*	Number of received control packets.
[inPktsDrop] **	*integer*	Number of received packets dropped for any reason. **Since: 4.28.2
inPktsLate **	*integer*	Number of received packets that failed sequence number check. **Since: 4.18.1
inPktsNoSCI	*integer*	Number of received packets dropped with unknown SCI.
inPktsNoTag	*integer*	Number of received packets without security tag when validateFrames is Strict.
inPktsNotUsingSA	*integer*	Number of received packets for an SA not in use.
inPktsNotValid	*integer*	Number of received invalid packets.
inPktsOK	*integer*	Number of received packets with no error.
inPktsTagged	*integer*	Number of received packets with security tag.
inPktsUnchecked	*integer*	Number of received packets not checked.
[inPktsUnknownSCI] **	*integer*	Number of received packets for an unknown SCI. **Since: 4.28.2

[inPktsUntagged] **	*integer*	Number of received packets without security tag when validateFrames is not Strict. **Since: 4.28.2
[inPktsUnusedSA] **	*integer*	Number of received packets for a SA not in use. **Since: 4.28.2
[inSecureChannelTcamHit] **	*integer*	Number of received packets that hit in the secure channel TCAM. **Since: 4.29.1
[inSecureChannelTcamMiss] **	*integer*	Number of received packets that missed the secure channel TCAM. **Since: 4.28.2
[inSubportTcamHit] **	*integer*	Number of received packets that hit in the subport TCAM. **Since: 4.29.1
[inSubportTcamMiss] **	*integer*	Number of received packets that missed the subport TCAM. **Since: 4.28.2
[inUcastPkts] **	*integer*	Number of received unicast packets on the controlled port. **Since: 4.28.2
[outBroadcastPkts] **	*integer*	Number of transmitted broadcast packets on the controlled port. **Since: 4.28.2
[outInvalidSA] **	*integer*	Number of transmitted packets which used an invalid SA. **Since: 4.29.1
[outMulticastPkts] **	*integer*	Number of transmitted multicast packets on the controlled port. **Since: 4.28.2
[outOctets] **	*integer*	Number of transmitted octets on the controlled port. **Since: 4.28.2
[outOctetsProtected] **	*integer*	Number of transmitted octets that were ICV protected but unencrypted. **Since: 4.29.1
outPktCtrl	*integer*	Number of transmitted control packets.
[outPktsProtected] **	*integer*	Number of transmitted packets that were ICV protected but unencrypted. **Since: 4.29.1
outPktsTooLong	*integer*	Number of transmitted packets larger than MTU.
outPktsUntagged	*integer*	Number of transmitted packets without security tag.
[outUcastPkts] **	*integer*	Number of transmitted unicast packets on the controlled port. **Since: 4.28.2

2.883 show mac security interface [...]

interfaces	*table(Interface name)* = *MacsecInterface* see 2.883.1	A mapping between interfaces and MAC security interface information.

2.883.1 MacsecInterface

address	*MAC address*	MAC address of the local port.
controlledPort	*boolean*	MAC security is operational.
[details]	*MacsecInterfaceDetailData* see 2.883.1.1	MAC security Interface detail information.
keyMsgId	*string*	Message identifier of the distributed key currently used.
keyNum	*integer*	Key number of the distributed key currently used.
portId	*integer*	Port number of the local port.

2.883.1.1 MacsecInterfaceDetailData

bypassProtocol **	*list of string*	List of protocols without MAC security protection. **Since: 4.22.1
cachedSak **	*boolean*	Cached SAK is used. **Since: 4.25.1
ethFlowControl **	*enum: Bypassed* | *Encrypted* | *None*	Flow control frame handling. **Since: 4.26.2
fipsPostStatus **	*enum: passed* | *none* | *inProgress* | *failed*	FIPS Power-on self-test. **Since: 4.21.3
keyServerPriority	*integer*	Configured key server priority for the interface.
latestKeyMsgId	*string*	A 96 bit message identifier for the latest key.
latestKeyMsgNum	*integer*	Message number of the latest key.
latestKeyReceiving	*boolean*	The latest key is currently used for decrypting data packets.
latestKeyTransmitting	*boolean*	The latest key is currently used for encrypting data packets.
localSsci	*string*	Value of local SSCI.
mkaLifeTime **	*integer*	Time in seconds after which peers are removed from live or potential peers list, upon no response. **Since: 4.26.0
oldKeyMsgId	*string*	A 96 bit message identifier for the old key.
oldKeyMsgNum	*integer*	Message number of the old key.
oldKeyReceiving	*boolean*	The old key is currently used for decrypting data packets.
oldKeyTransmitting	*boolean*	The old key is currently used for encrypting data packets.
profileName **	*string*	MAC security profile name. **Since: 4.24.1

ptpBypass **	*list of MacsecPtpBypassSupport* see 2.883.1.1.1	List of supported PTP bypasses. **Since: 4.29.2
sakRetireDelay **	*integer*	Time in seconds after which the old SAK is retired. **Since: 4.29.0
sakTransmitDelay **	*integer*	Time in seconds after which the latest key transmit is enabled on key server. **Since: 4.29.0
sessionReKeyPeriod	*floating point number*	Period in seconds after which the session keys are refreshed.
[suspensionEvent] **	*enum: sso* | *suspended*	Event due to which MKA is suspended. **Since: 4.30.0
traffic **	*enum: Protected* | *Unprotected* | *None*	Type of traffic. **Since: 4.21.3
unauthBypassProtocol **	*list of string*	List of protocols allowed in unauthorized state. **Since: 4.29.2

2.883.1.1.1 MacsecPtpBypassSupport

support	*enum: multicast* | *dataPlane* | *controlPlane* | *notSupported* | *unicast* | *full*	Supported PTP bypasses.

2.884 show mac security mka counters [...]

interfaces	*table(Interface name)* = *MacsecMessageCounters-Interface* see 2.884.1	A mapping between interfaces and MAC security MKA counters.

2.884.1 MacsecMessageCountersInterface

[details]	*MacsecMessageCounter-Detail* see 2.884.1.1	MAC security counter detailed information.
rxPacketsFailure	*integer*	Total of invalid/errored received packets.
rxPacketsSuccess	*integer*	Total of successfully received packets.
txPacketsFailure	*integer*	Total of invalid/errored transmitted packets.
txPacketsSuccess	*integer*	Total of successfully transmitted packets.

2.884.1.1 MacsecMessageCounterDetail

rxBasicParamSetError	*integer*	Basic parameter set errored received packets.
rxDistCakSetError **	*integer*	Dist CAK Set errored received packets. **Since: 4.20.5
rxDistSakSetError	*integer*	Dist SAK Set errored received packets.
rxEapolError	*integer*	EAPOL errored packets.
rxIcvIndicatorError **	*integer*	ICV Indicator errored received packets. **Since: 4.18.1
rxIcvValidationError	*integer*	ICV validation error in received packets.
rxInvalid	*integer*	Invalid interface or null packets.
rxLivePeerListError	*integer*	Live peerlist set errored received packets.
rxPotentialPeerListError	*integer*	Potential peerlist set errored received packets.
rxSakUseSetError	*integer*	SAK Use Set errored received packets.
rxUnrecognizedCkn	*integer*	Unrecognized ckn in received packets.
rxUnrecognizedSetError	*integer*	Unrecognized parameter set errored received packets.
rxXpnSetError **	*integer*	XPN Set errored received packets. **Since: 4.30.0
txInvalid	*integer*	Invalid interface in tx packets.

2.885 show mac security participants [...]

interfaces	*table(Interface name) = MacsecParticipants-Interface* see 2.885.1	A mapping between interfaces and MAC security interface participants.

2.885.1 MacsecParticipantsInterface

participants	*table(string) = MacsecParticipant* see 2.885.1.1	A mapping between CKN and MAC security participants.

2.885.1.1 MacsecParticipant

defaultActor	*boolean*	The participant is spawned from a configured fallback key.
[details]	*MacsecParticipantDetail* see 2.885.1.1.1	MAC security participant detail information.
electedSelf	*boolean*	The participant has elected itself as the key server.
msgId	*string*	Member identifier generated for the participant.
principalActor	*boolean*	The participant is also a principal actor.
success	*boolean*	The participant has successfully elected a key server and has at least one live peer.

2.885.1.1.1 MacsecParticipantDetail

keyNum	*integer*	Key number of the distributed session association key.
keyServerAddr	*MAC address*	Mac address of the key server.
keyServerMsgId	*string*	Message identifier of the key server.
keyServerPortId	*integer*	Port Id of the key server.
livePeerList	*list of string*	List of all live peers for this participant.
llpnExhaustion	*integer*	Number of times LLPN exhaustion event detected when this participant is the principal actor.
potentialPeerList	*list of string*	List of all potential peers for this participant.
sakTransmit	*boolean*	The participant is using the distributed SAK for transmit.

2.886 show mac security profile [...]

sources	*table(string) = MacsecProfiles* see 2.886.1	Mapping between source and profiles.

2.886.1 MacsecProfiles

profiles	*table(string) = MacsecProfile* see 2.886.1.1	Mapping between profile name and profile details.

2.886.1.1 MacsecProfile

bypassProtocol **	*list of string*	List of protocols without MAC security protection. **Since: 4.28.1
cakSha256Hash	*string*	Sha256 hash of primary connectivity association key.
cipher	*enum: aes128-gcm | aes256-gcm-xpn | aes256-gcm | aes128-gcm-xpn*	Cipher suite for a profile.
ckn	*string*	Primary connectivity association key name.
fallbackCakSha256Hash	*string*	Sha256 hash of fallback connectivity association key.
fallbackCkn	*string*	Fallback connectivity association key name.
[fallbackKeySource] **	*enum: sharedSecretProfile | dot1x | preSharedKey*	Pre-shared CAK/CKN, 802.1x derived CAK/CKN or CAK/CKN from a shared-secret profile. **Since: 4.27.0
fallbackSecProfName **	*string*	Fallback shared secret profile name. **Since: 4.27.2
includeSci **	*boolean*	Include secure channel identifier in data packets. **Since: 4.28.1
interfaces	*list of Interface name*	List of interfaces this profile is configured on.
keyRetirePolicy	*enum: delayed | immediate*	Key retirement policy.
keyServerPriority **	*integer*	Configured key server priority for the interface. **Since: 4.28.1
mkaLifeTime **	*integer*	Time in seconds after which peers are removed from live or potential peers list, upon no response. **Since: 4.26.0
[primaryKeySource] **	*enum: sharedSecretProfile | dot1x | preSharedKey*	Pre-shared CAK/CKN, 802.1x derived CAK/CKN or CAK/CKN from a shared-secret profile. **Since: 4.27.0
priority	*integer*	MACsec config priority.

sakAnMax **	*integer*	Maximum value of AN of any SAK generated when the interface configured with this profile is keyserver. **Since: 4.28.2
sakRetireDelay **	*integer*	Time in seconds after which the old SAK is retired. **Since: 4.29.0
sakTransmitDelay **	*integer*	Time in seconds after which the latest key transmit is enabled on key server. **Since: 4.29.0
secProfName **	*string*	Primary shared secret profile name. **Since: 4.27.2
sessionReKeyPeriod **	*integer*	Period in seconds after which the session keys are refreshed. **Since: 4.28.1
source	*string*	Cli or EosSdk agent name.
unauthBypassProtocol **	*list of string*	List of protocols allowed in unauthorized state. **Since: 4.29.2
unprotectedTrafficPolicy **	*enum: drop | allow | allow active-sak*	Unprotected traffic policy. **Since: 4.25.1

2.887 show mac security ptp bypass

interfaces	*table(Interface name) = MacsecPtpBypassInterface* see 2.887.1	A mapping between interfaces and PTP bypass information.

2.887.1 MacsecPtpBypassInterface

bypassSupport	*list of MacsecPtpBypassSupport* see 2.883.1.1.1	List of supported PTP bypasses.

2.888 show mac security status

activeProfiles	*integer*	Number of profiles configured in at least-one interface.
delayProtection **	*boolean*	Data delay protection is enabled. **Since: 4.20.5
eapolDestMac **	*MAC address*	MACsec EAPoL destination MAC address. **Since: 4.25.0
fipsMode	*boolean*	FIPS Mode is enabled.
[hwLicenseEnabled] **	*boolean*	Hardware license is enabled. **Since: 4.25.2
licenseEnabled **	*boolean*	License is enabled. **Since: 4.21.3
securedInterfaces	*integer*	Number of interfaces with MAC security enabled.

2.889 show mac source-check invalid

sourceCheckInvalid	*boolean*	Source check invalid Mac.

2.890 show mac source-check system-mac

sourceCheckCPUMac	*boolean*	Source check CPU Mac.

2.891 show mac timestamp keyframe profile [...]

profiles	*table(string) = Profile* see 2.891.1	Keyframe profiles configuration, keyed by name.

2.891.1 Profile

[egressInterfaces]	*list of Interface name*	Egress interfaces with this profile applied.
[profileConfig]	*ProfileConfig* see 2.891.1.1	Detailed configuration of the profile.

2.891.1.1 ProfileConfig

clockSkew	*boolean*	Clock skew is appended to the keyframe.
destinationIp	*IPv4 address*	Destination IPv4 address.
destinationMac	*MAC address*	Destination MAC address.
deviceId	*integer*	Device identifier.
hasDefaultSourceIp	*boolean*	The source IPv4 address is the default one.
interval	*integer*	Number of milliseconds between each keyframe.
sourceIp	*IPv4 address*	Source IPv4 address.
txDropCount	*boolean*	TxDropCount is appended to the keyframe.
vlanId	*integer*	Identifier for a Virtual LAN.

2.892 show mac traffic-policy field-set [...]

sources	*table(string) = MacTrafficPolicyMacFieldSetSource* see 2.892.1	Mapping of source name to MAC field set configuration.

2.892.1 MacTrafficPolicyMacFieldSetSource

macFieldSets	*table(string) = MacTrafficPolicyMacFieldSet* see 2.892.1.1	Mapping of field set name to MAC address information.

2.892.1.1 MacTrafficPolicyMacFieldSet

macAddrs	*list of MAC address*	List of all MAC addresses.

2.893 show mac traffic-policy summary

sources	*table(string) = MacTrafficPolicySourceSummary* see 2.893.1	A mapping of source path to its configuration.

2.893.1 MacTrafficPolicySourceSummary

enabled	*boolean*	Enable status of MAC traffic policy.
rules	*table(string) = RuleSummary* see 2.893.1.1	A mapping of rule to its configuration.

2.893.1.1 RuleSummary

actions	*RuleAction* see 2.893.1.1.1	Actions of the rule.
matches	*RuleMatch* see 2.893.1.1.2	Match criteria for the rule.

2.893.1.1.1 RuleAction

[accessList]	*AccessListAction* see 2.893.1.1.1.1	Access list information.

2.893.1.1.1.1 AccessListAction

name	*string*	Access list name.

2.893.1.1.2 RuleMatch

[macFieldSet]	*MacFieldSet* see 2.893.1.1.2.1	MAC field set information.

2.893.1.1.2.1 MacFieldSet

name	*string*	Name of MAC field set.

2.894 show maintenance

interfaces	*table(Interface name) = MaintenanceStatus* see 2.894.1.1	Map Interfaces to their Maintenance status.
units	*table(string) = MaintenanceStatus* see 2.894.1.1	Map Unit name to status.
vrfs	*table(string) = MaintenanceProtocolInfo* see 2.894.1	Map Vrf name to its members and their Maintenance status.

2.894.1 MaintenanceProtocolInfo

bgpPeers	*table(string) = MaintenanceStatus* see 2.894.1.1	Map Bgp Peers to their Maintenance status.

2.894.1.1 MaintenanceStatus

adminState **	*enum: underMaintenance | active*	Admin state of unit under Maintenance. **Since: 4.18.0
[aggInBpsRate] **	*integer*	Aggregate input bps rate of the Unit. **Since: 4.18.1
[aggOutBpsRate] **	*integer*	Aggregate output bps rate of the Unit. **Since: 4.18.1
[intfsViolatingTraffic-Threshold]	*boolean*	Interfaces part of this unit are violating traffic threshold after reaching maintenance.
[onBootMaintenance]	*boolean*	Maintenance operation due to on-boot config.
state	*enum: failedEnter | maintenanceModeExit | failedExit | maintenanceModeEnter | underMaintenance | active | init*	Operational state of unit under Maintenance.
stateChangeTime	*floating point number*	UTC time at which the operational state of Maintenance unit changed.

2.895 show maintenance bgp (IP_PEER | IP6_PEER | IP6_LL_PEER) [...]

revision 5

vrfs	*table(string) = BgpPeerMaintenanceInfo* see 2.895.1	Per VRF BGP peer maintenance information.

2.895.1 BgpPeerMaintenanceInfo

groups	*list of string*	List of Groups that the peer is part of.
peer	*string*	Neighbor address.
routeMapInfo	*RouteMaps* see 2.895.1.2	Route map information.
routeMapInfoIn **	*RouteMaps* see 2.895.1.2	Inbound route map information. **Since: 4.24.0
routeMapInfoOut **	*RouteMaps* see 2.895.1.2	Outbound route map information. **Since: 4.24.0
[selectedProfile]	*string*	Selected profile from BGP groups.
status	*BgpMaintenanceStatusPerPeer* see 2.895.1.1	BGP Maintenance status of this peer.

2.895.1.1 BgpMaintenanceStatusPerPeer

asn	*integer*	Autonomous System Number.
peers	*table(string) = BgpPeerMaintenanceStatus* see 2.895.1.1.1	Dictionary of BGP peer entries indexed by the peer address.
routerId	*IPv4 address*	BGP Router Identity.
vrf	*string*	VRF Name.

2.895.1.1.1 BgpPeerMaintenanceStatus

[appliedRouteMap]	*string*	Maintenance inout route map applied for the peer.
[appliedRouteMapIn] **	*string*	Maintenance inbound route map applied for the peer. **Since: 4.24.0
[appliedRouteMapOut] **	*string*	Maintenance outbound route map applied for the peer. **Since: 4.24.0
underMaintenance	*boolean*	Peering under maintenance.

2.895.1.2 RouteMaps

Could also be an instance of:
ConfigSanityRouteMaps 2.895.1.3

routeMaps	*table(string) = RouteMap* see 2.895.1.2.1	Route maps by name.

2.895.1.2.1 RouteMap

entries	*table(numeric string) = RouteMapEntry* see 2.895.1.2.1.1	Route map entries by sequence number.

2.895.1.2.1.1 RouteMapEntry

[continueClause]	*boolean*	Continue to next clause.
[continueSeqno]	*integer*	Sequence number of next clause.
description	*list of string*	Route map entry description.
filterType	*enum: permit | deny*	Filter type.
matchRules	*MatchRules* see 2.127.1.3.2.1	Match rules.
setRules	*SetRules* see 2.127.1.3.2.8	Set rules.
[subRouteMap]	*SubRouteMap* see 2.895.1.2.1.2	Sub route map.

2.895.1.2.1.2 SubRouteMap

[debugInfo]	*SubRMDebugInfo* see 2.895.1.2.1.3	Debugging information for sub-route-maps.
[invert]	*boolean*	Invert sub route map result.
name	*string*	Sub route map name.

2.895.1.2.1.3 SubRMDebugInfo

[name]	*AttributeDebug* see 2.127.1.3.2.6	Debugging information for sub-route-maps.

2.895.1.3 ConfigSanityRouteMaps

routeMaps	*table(string) = RouteMap* see 2.895.1.2.1	Route maps by name.

2.896 show maintenance bgp ip all [...]

revision 2

vrfs	*table(string) = BgpMaintenanceStatus* see 2.896.1	Per VRF BGP peer maintenance status information.

2.896.1 BgpMaintenanceStatus

asn	*integer*	Autonomous System Number.
peers	*table(string) = BgpPeerMaintenanceStatus* see 2.895.1.1.1	Dictionary of BGP peer entries indexed by the peer address.
routerId	*IPv4 address*	BGP Router Identity.

vrf	*string*	VRF Name.

2.897 show maintenance bgp ipv6 all [...]

revision 2

see 2.896

2.898 show maintenance bgp receiver route-map

revision 5

see 2.895.1.2

2.899 show maintenance debug [...]

maintenanceEventInfo	*table(string) = MaintenanceEventInfo* see 2.899.1	Map Unit name to historical information.

2.899.1 MaintenanceEventInfo

maintEnterEvents	*list of MaintenanceEventInfoEntry* see 2.899.1.1	Historical information for maintenance enter of unit.
maintExitEvents	*list of MaintenanceEventInfoEntry* see 2.899.1.1	Historical information for maintenance exit of unit.

2.899.1.1 MaintenanceEventInfoEntry

eventInfo	*string*	Information about a Maintenance related event.
timeStamp	*floating point number*	Time at which the event occured.

2.900 show maintenance debug bgp

see 2.899

2.901 show maintenance debug bgp (GROUP | IP_PEER | IP6_PEER | IP6_LL_PEER) [...]

see 2.899

2.902 show maintenance groups

bgpGroups	*table(string) = BgpGroup* see 2.902.1	Map group name to BGP Group information.
intfGroups	*table(string) = InterfaceGroup* see 2.902.2	Map group name to Interface group information.

2.902.1 BgpGroup

[bgpProfile]	*string*	Bgp profile associated with group.
neighbors	*table(string) = BgpNeighbors* see 2.902.1.1	Map Peer Type to list of Peers, where peer type, the key, could be either peerGroup, peerIpv4 or peerIpv6.
origin	*enum: builtin | userConfigured*	origin of this group.
units	*list of string*	List of Maintenance units of which this group is a member of.
vrfName	*string*	Name of the Vrf.

2.902.1.1 BgpNeighbors

peers	*list of string*	List of peers, which could be either peer groups, IPv4 or IPv6 peers.

2.902.2 InterfaceGroup

bgpProfile	*string*	BGP Maintenance Profile for this group.
interfaces	*list of string*	List of Interfaces in the group.
intfProfile	*string*	Interface Maintenance Profile for this group.
origin	*enum: builtin | userConfigured*	origin of this group.
units	*list of string*	List of Maintenance Units of which this group is a member of.

2.903 show maintenance groups bgp [...]

bgpGroups	*table(string) = BgpGroup* see 2.902.1	Map group name to BGP Group information.

2.904 show maintenance groups interface [...]

intfGroups	*table(string) = InterfaceGroup* see 2.902.2	Map group name to Interface Group information.

2.905 show maintenance interface [...]

interfaces	*table(Interface name)* = *MaintenanceInterfaceBrief* see 2.905.1	Maintenance mode information for interfaces.

2.905.1 MaintenanceInterfaceBrief

adminState **	*enum: underMaintenance* | *active*	Maintenance admin state of this interface. **Since: 4.18.0
[lastInBpsRate]	*floating point number*	Input bps rate.
[lastOutBpsRate]	*floating point number*	Output bps rate.
[maintDownTime]	*floating point number*	Time at which the interface was shutdown for maintenance.
state	*enum: failedEnter* | *maintenanceModeExit* | *failedExit* | *maintenanceModeEnter* | *underMaintenance* | *active* | *init*	Maintenance operational state of this interface.
[trafficAboveThreshold]	*boolean*	Indicates if the traffic on the interface is above threshold.

2.906 show maintenance interface detail [...]

revision 2

interfaces	*table(Interface name)* = *MaintenanceInterfaceStatus* see 2.906.1	Maintenance mode information for interfaces.

2.906.1 MaintenanceInterfaceStatus

adminState	*enum: underMaintenance* | *active*	Maintenance admin state of this interface.
[bgpProfile]	*string*	Name of the BGP Maintenance Profile applied during Maintenance.
[bgpStatus]	*MaintenanceInterfaceBgpStatus* see 2.906.1.1	BGP status for the interface.
groups	*list of string*	Groups that the interface is part of.
history	*list of string*	Historical information for dynamic interface unit.
[intfProfile]	*string*	Name of the Interface Maintenance Profile applied during Maintenance.
[rateMonStatus]	*MaintenanceInterfaceRateMonStatus* see 2.906.1.2	Rate Monitoring status for the interface.
[state]	*enum: failedEnter* | *maintenanceModeExit* | *failedExit* | *maintenanceModeEnter* | *underMaintenance* | *active* | *init*	Maintenance operational state of this interface.

2.906.1.1 MaintenanceInterfaceBgpStatus

maintState	*enum: failedEnter* | *maintenanceModeExit* | *failedExit* | *maintenanceModeEnter* | *underMaintenance* | *active* | *init*	Maintenance State of the Interface.
peers	*table(string)* = *MaintenanceInterfaceBgpPeerStatus* see 2.906.1.1.1	Maintenance state of the peers on this interface.
vrf	*string*	VRF Name.

2.906.1.1.1 MaintenanceInterfaceBgpPeerStatus

[appliedRouteMap]	*string*	Maintenance routemap applied for the peer.
[appliedRouteMapIn] **	*string*	Maintenance inbound routemap applied for the peer. **Since: 4.24.0

[appliedRouteMapOut] **	*string*	Maintenance outbound routemap applied for the peer. **Since: 4.24.0

2.906.1.2 MaintenanceInterfaceRateMonStatus

lastInBpsRate	*floating point number*	Input bps rate.
lastOutBpsRate	*floating point number*	Output bps rate.
lastSampleTime	*floating point number*	The time at which the last sample was taken.
[maintDownTime]	*floating point number*	Time at which the interface was shutdown for maintenance.
numSamplesAboveThreshold-PostMaint	*integer*	Number of samples above threshold, after maintenance.
numSamplesAboveThreshold-PreMaint	*integer*	Number of samples above threshold, before maintenance.
numSamplesBelowThreshold-PostMaint	*integer*	Number of samples below threshold, after maintenance.
numSamplesBelowThreshold-PreMaint	*integer*	Number of samples below threshold, before maintenance.
profile	*string*	Name of the Interface profile being used to monitor the interface.
startTime	*floating point number*	The time at which rate monitoring was started for this interface.
state	*enum: monitorStop | monitorPassive | monitorInit | monitorThreshold*	State of traffic rate monitoring for the interface.
stateTime	*floating point number*	The time at which the state was entered.
totalSamples	*integer*	Total number of samples collected so far.

2.907 show maintenance profiles

bgpProfiles	*table(string) = BgpProfile* see 2.907.1	Map profile name to BGP Profile information.
intfProfiles	*table(string) = InterfaceProfile* see 2.907.2	Map profile name to Interface Profile information.
unitProfiles	*table(string) = UnitProfile* see 2.907.3	Map profile name to Unit Profile information.

2.907.1 BgpProfile

routeMapIn **	*string*	Inbound initiator route map for BGP profile. **Since: 4.24.0
routeMapInOut	*string*	Initiator route map for BGP profile.
routeMapOut **	*string*	Outbound initiator route map for BGP profile. **Since: 4.24.0

2.907.2 InterfaceProfile

rateMonLoadInterval	*integer*	Duration, in seconds, for which the traffic needs to remain below the threshold before the interface can be declared to be under maintenance.
rateMonThreshold	*floating point number*	Traffic threshold, in Kbps.
shutdown	*boolean*	Whether to shutdown interface when the traffic goes below threshold.
shutdownMaxDelay	*floating point number*	Maximum time to wait, in seconds, before shutting down the interface, if shutdown is configured.

2.907.3 UnitProfile

onBootDuration	*integer*	Duration, in seconds, for which the unit will be under maintenance on boot.
onBootEnabled	*boolean*	Whether onboot is enabled or not.

2.908 show maintenance profiles bgp [...]

bgpProfiles	*table(string) = BgpProfile* see 2.907.1	Map BGP Profile name to BGP Profile information.

2.909 show maintenance profiles bgp default

revision 5

profileName	*string*	Profile name.
[routeMap]	*string*	Default route map name.
[routeMapIn] **	*string*	Default inbound route map name. **Since: 4.24.0
routeMapInfo	*RouteMaps* see 2.895.1.2	Route map information.
routeMapInfoIn **	*RouteMaps* see 2.895.1.2	Inbound route map information. **Since: 4.24.0
routeMapInfoOut **	*RouteMaps* see 2.895.1.2	Outbound route map information. **Since: 4.24.0
[routeMapOut] **	*string*	Default outbound route map name. **Since: 4.24.0

2.910 show maintenance profiles interface [...]

intfProfiles	*table(string) = InterfaceProfile* see 2.907.2	Map Interface Profile name to Interface Profile information.

2.911 show maintenance profiles interface default

[profileAttributes]	*InterfaceProfile* see 2.907.2	Attributes of the default Interface profile.
profileName	*string*	Default Interface Profile Name.

2.912 show maintenance profiles unit [...]

unitProfiles	*table(string) = UnitProfile* see 2.907.3	Map Unit Profile name to Unit Profile.

2.913 show maintenance profiles unit default

[profileAttributes]	*UnitProfile* see 2.907.3	Attributes of the default Unit profile.
profileName	*string*	Default Unit Profile Name.

2.914 show maintenance stages [...]

maintenanceEnterStages	*list of MaintenanceStage* see 2.914.1	List of Maintenance enter stages.
maintenanceExitStages	*list of MaintenanceStage* see 2.914.1	List of Maintenance exit stages.

2.914.1 MaintenanceStage

stageDescription	*string*	Maintenance stage description.
stageName	*string*	Maintenance stage name.

2.915 show maintenance summary

intfDirectlyInMaintenance	*table(string) = integer*	Map Maintenance States to number of Interfaces directly put in Maintenance.
numUnitsConfigured	*integer*	Number of Units configured.
peerDirectlyInMaintenance	*table(string) = integer*	Map Maintenance States to number of Peers directly put in Maintenance.
[rateMonSummary]	*MaintenanceRateMonSummary* see 2.915.1	Interface Rate Monitoring summary.
unitStatistics	*table(string) = integer*	Map different Maintenance States to number of Units.

2.915.1 MaintenanceRateMonSummary

numIntfsEnterMaint	*integer*	Number of interfaces entering maintenance.
numIntfsMaintDown	*integer*	Number of interfaces shutdown for maintenance.
numIntfsThresholdViolation	*integer*	Number of interfaces which violated the traffic threshold after going under maintenance.
numIntfsUnderMaint	*integer*	Number of interfaces under maintenance.

2.916 show maintenance units [...]

maintenanceUnits	*table(string) = MaintenanceUnit* see 2.916.1	Map Unit name to Unit.

2.916.1 MaintenanceUnit

adminState **	*enum: underMaintenance | active*	Admin state of the Maintenance Unit. **Since: 4.18.0
bgpGroups	*list of string*	List of Bgp Groups.
history	*list of MaintenanceEventInfoEntry* see 2.899.1.1	Historical information for unit.
interfaceGroups	*list of string*	List of Interface Groups.
[intfsViolatingTraffic-Threshold]	*list of Interface name*	List of interfaces which are violating traffic threshold.
[onBootExitTime]	*floating point number*	Exit time of on-boot maintenance.
[onBootMaintenance]	*boolean*	Maintenance operation due to on-boot config.
[origin]	*enum: builtin | userConfigured*	origin of this unit.
state	*enum: failedEnter | maintenanceModeExit | failedExit | maintenanceModeEnter | underMaintenance | active | init*	Operational state of the Maintenance Unit.
[timeOfStateChange]	*floating point number*	Time since last state change.
[totalIntfTraffic-Violations]	*integer*	Number of interfaces which have violated traffic threshold since maintenance.
unitProfile	*string*	Name of the Unit Maintenance Profile applied during Maintenance.

2.917 show management api gnmi

revision 2

enabled	*boolean*	Aggregate transports enabled.
grpctunnelClients **	*table(string) = GrpctunnelClientStatus* see 2.917.2	gRPC-tunnel clients indexed by name. **Since: 4.27.0
octa	*boolean*	Octa enabled.
setPersistence **	*boolean*	Set is persistent. **Since: 4.27.2
transports	*table(string) = GnmiEndpointStatus* see 2.917.1	Transports indexed by name.

2.917.1 GnmiEndpointStatus

accounting **	*boolean*	RPC requests accounting. **Since: 4.27.2
authorization	*boolean*	per-RPC authorization.
certUsernameAuthn **	*boolean*	Certificate username authentication. **Since: 4.27.1
enabled	*boolean*	Endpoint is enabled.
error	*string*	error which occurred while starting the endpoint.
listenAddrs **	*list of IpGenericAddr* see 2.296.1.1.1.1	Addresses for gRPC to listen on. **Since: 4.27.0
port	*integer*	The port this protocol's server is listening on.
qosDscp	*integer*	The QoS DSCP value.
sslProfile	*string*	SSL profile name.
useNotifSendTime **	*boolean*	Subscription notification timestamp. **Since: 4.27.0
vrfName	*string*	The VRF this protocol's server is listening in.

2.917.2 GrpctunnelClientStatus

destinationHost	*string*	Destination address the client is dialing to.
destinationPort	*integer*	Destination port the client is dialing to.
enabled	*boolean*	gRPC-tunnel dial-out client is enabled.
error	*string*	Error which occurred while starting the gRPC-tunnel dial-out client.
gnmiSslProfile	*string*	SSL profile used by the gNMI server.
localAddressAndPort	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Local address/port the client is dialing from.
localInterface	*interface name*	Local interface the client is dialing from.
status	*string*	Status of the gRPC-tunnel dial-out client.
targetId	*string*	Target ID used by the gRPC-tunnel connection.
tunnelSslProfile	*string*	SSL profile used by the tunnel.

`vrf`	*string*	VRF of the gRPC-tunnel dial-out client.

2.918 show management api gnmi access-list [...]

revision 2

`transports`	*table(string)* = *AllAclList* see 2.53	Transport ACLs indexed by transport name.

2.919 show management api gribi

certUsernameAuthn **	*boolean*	Use the username from the client TLS certificate for AAA authentication. **Since: 4.28.1
dscp **	*integer*	The DSCP value set for outgoing packets. **Since: 4.27.2
enabled	*boolean*	The endpoint is enabled.
[error]	*string*	Error which occurred while starting the endpoint.
port	*integer*	The port this protocol's server is listening on.
[rpcStatus] **	*GribiRpcStatus* see 2.919.1	gRIBI RPC server and client status. **Since: 4.27.0
[serviceAcl] **	*string*	The access control group to filter which gRIBI clients can connect via IPv4. **Since: 4.28.0
[serviceAclV6] **	*string*	The access control group to filter which gRIBI clients can connect via IPv6. **Since: 4.28.1
[sslProfile]	*string*	SSL profile name.
started **	*boolean*	gRIBI server is started. **Since: 4.27.0
vrfName	*string*	The VRF this protocol's server is listening in.

2.919.1 GribiRpcStatus

highestElectionId	*GribiElectionId* see 2.919.1.1	Highest election ID received so far.
highestElectionIdTimeStamp	*floating point number*	Timestamp when highest election ID is received.
modifyRpcClients	*table(IPv4 or IPv6 address) = GribiModifyRpcClientIpStatus* see 2.919.1.2	Map of Modify RPC client status, keyed by client IP.
primaryClientId	*integer*	Current Modify RPC primary client internal ID.
[primaryClientIp]	*IP address*	Current Modify RPC primary client IP address.
[primaryClientPort]	*integer*	Current Modify RPC primary client port.
totalClients	*integer*	Total number of Modify RPC client connections.

2.919.1.1 GribiElectionId

high	*integer*	High 64 bits of the 128-bit election ID.
low	*integer*	Low 64 bits of the 128-bit election ID.

2.919.1.2 GribiModifyRpcClientIpStatus

ports	*table(numeric string) = GribiModifyRpcClientIpPortStatus* see 2.919.1.2.1	Map of Modify RPC status, keyed by port.

2.919.1.2.1 GribiModifyRpcClientIpPortStatus

streams	*table(numeric string) = GribiModifyRpcStreamStatus* see 2.919.1.2.1.1	Map of Modify RPC stream status, keyed by stream ID.

2.919.1.2.1.1 GribiModifyRpcStreamStatus

[aftOpAckTypeApplied]	*enum: ribAck | invalid | ribAndFibAck*	AFT operation acknowledgement type applied.
[aftOpAckTypeRequested]	*enum: ribAck | invalid | ribAndFibAck*	AFT operation acknowledgement type requested.
[aftPersistence]	*enum: delAft | invalid | preserve*	AFT persistence.
[clientRedundancy]	*enum: allPrimary | singlePrimary | invalid*	Client redundancy.
connectionTimeStamp	*floating point number*	Timestamp when the stream connected.
[electionId]	*GribiElectionId* see 2.919.1.1	Election ID received.
[electionIdTimeStamp]	*floating point number*	Timestamp when election ID is received.
lastAftOpTimeStamp	*floating point number*	Timestamp of last AFT operation received.
sessionParamsTimeStamp	*floating point number*	Timestamp when session parameters are received.

2.920 show management api gribi access-list [...]

transports	*table(string) = AllAclList* see 2.53	Transport ACLs indexed by transport name.

2.921 show management api http-commands

bytesIn	*integer*	The number of bytes received from HTTP requests.
bytesOut	*integer*	The number of bytes sent in HTTP responses.
commandCount	*integer*	The number of CLI commands requested to run.
[corsOrigins]	*list of string*	Allowed CORS Origins. Present when CORS is supported.
[dscpValue]	*integer*	The QoS DSCP value set in outgoing IP packets.
enabled	*boolean*	Whether the server is globally enabled.
executionTime	*floating point number*	The time in seconds spent executing CLI commands.
[fipsEnabled]	*boolean*	Whether FIPS mode is enabled for HTTPS algorithms.
hitCount	*integer*	The number of incoming HTTP requests the server has seen.
httpServer	*HttpProtocolStatus* see 2.921.1	The status of the HTTP server.
httpsServer	*HttpProtocolStatus* see 2.921.1	The status of the HTTPS server.
iframeAncestors	*list of string*	Allowed Iframe ancestors.
[lastHitTime]	*floating point number*	The UTC timestamp of the last request.
[localHttpServer]	*HttpProtocolStatus* see 2.921.1	The status of the localhost HTTP server.
logLevel **	*enum: warn | error | emerg | info | debug | crit | notice | alert | none*	Logging level for the HTTP server. **Since: 4.20.0
requestCount	*integer*	The number of valid JSON-RPC requests received.
[sslProfile]	*SslProfile* see 2.921.2	The configuration and status of the SSL profile.
tlsProtocol	*list of string*	Allowed TLS protocol versions.
[unixSocketServer]	*UnixProtocolStatus* see 2.921.3	The status of the Unix Domain Socket server.
urls	*list of string*	URLs the server is listening on.
users	*table(string) = UserStatistics* see 2.921.4	Mapping from username to statistics about that user.
vrf	*string*	The VRF the API is configured to serve in.
vrfs **	*list of string*	All VRFs enabled in CAPI. **Since: 4.20.0

2.921.1 HttpProtocolStatus

configured	*boolean*	Whether this protocol is configured to be enabled.
port	*integer*	The port this protocol's server is listening on.

running	*boolean*	Whether this protocol is currently serving and accepting requests.

2.921.2 SslProfile

configured	*boolean*	Whether this profile is configured in SSL Manager.
name	*string*	SSL profile name.
[state]	*enum: valid | invalid | updating*	State of the SSL profile.

2.921.3 UnixProtocolStatus

configured	*boolean*	Whether this protocol is configured to be enabled.
running	*boolean*	Whether this protocol is currently serving and accepting requests.

2.921.4 UserStatistics

bytesIn	*integer*	The number of bytes received from the user's requests.
bytesOut	*integer*	The number of bytes sent in responses to the user.
lastHitTime	*floating point number*	The UTC timestamp of the user's last request.
requestCount	*integer*	Number of requests received from the user.

2.922 show management api http-commands ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.923 show management api http-commands https certificate

revision 2

certificate	*string*	Certificate used by SSL/TLS.

2.924 show management api http-commands https cipher

enabledCipherSuites	*list of CipherSuite* see 2.924.1	Enabled CipherSuites.

2.924.1 CipherSuite

`cipherSuite`	*enum: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA | TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA | TLS_PSK_WITH_RC4_128_SHA | TLS_DHE_RSA_WITH_AES_128_CBC_SHA | TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 | TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA | TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 | TLS_RSA_WITH_NULL_MD5 | TLS_DHE_DSS_WITH_SEED_CBC_SHA | TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 | TLS_ECDH_RSA_WITH_AES_256_CBC_SHA | TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 | TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 | TLS_DH_DSS_WITH_AES_128_CBC_SHA | TLS_ECDHE_ECDSA_WITH_RC4_128_SHA | TLS_RSA_WITH_IDEA_CBC_SHA | TLS_KRB5_WITH_IDEA_CBC_MD5 | TLS_RSA_WITH_RC4_128_SHA | TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA | TLS_ECDH_anon_WITH_AES_128_CBC_SHA | TLS_PSK_WITH_AES_128_CBC_SHA | TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 | TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA | TLS_DHE_DSS_WITH_AES_128_CBC_SHA | TLS_DHE_RSA_WITH_AES_256_CBC_SHA | TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 | TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA | TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA |*	CipherSuite Name.

2.925 show management api http-commands https diffie-hellman

`diffieHellmanParameters`	*string*	Parameters used by SSL/TLS for diffie-hellman key exchange.

2.926 show management api models

revision 2

aftIPv4Unicast	*boolean*	IPv4 Unicast is enabled for AFT.
aftIPv6Unicast	*boolean*	IPv6 Unicast is enabled for AFT.
aftMpls **	*boolean*	MPLS is enabled for AFT. **Since: 4.28.1
aftRouteSummary **	*boolean*	OpenConfig route summary is enabled for AFT. **Since: 4.30.1
disabledSmashPaths	*list of string*	List of disabled Smash paths.
disabledSysdbPaths	*table(string) = SysdbExcludesStatus* see 2.926.4	A mapping of a disabled Sysdb path to its status.
enabledSmashPaths	*list of string*	List of enabled Smash paths.
ipfixOptions **	*IpfixOptions* see 2.926.1	Set of IPFIX options. **Since: 4.26.2
isisOptions **	*IsisOptions* see 2.926.2	Set of IS-IS options. **Since: 4.28.2
sflowOptions **	*SflowOptions* see 2.926.3	Set of sFlow options. **Since: 4.26.2
yangPaths **	*table(string) = boolean*	A mapping of a YANG path to whether it is explicitly enabled. **Since: 4.27.2

2.926.1 IpfixOptions

[address]	*string*	Listening address for the collector.
[domain]	*string*	Domain value used by the collector.
enableCollector	*boolean*	The collector is enabled.
[flowLifetime]	*integer*	Lifetime (in seconds) of entries in internal flow table.
[flowTableSize]	*integer*	Maximum number of entries in internal flow table.
[port]	*integer*	Listening port for the collector.

2.926.2 IsisOptions

linkStateDatabase	*boolean*	Set if Link State Database is enabled.

2.926.3 SflowOptions

[address]	*string*	Listening address for the collector.
[domain]	*string*	Domain value used by the collector.
enableCollector	*boolean*	The collector is enabled.
[flowLifetime]	*integer*	Lifetime (in seconds) of entries in internal sample table.
[flowTableSize]	*integer*	Maximum number of entries in internal sample table.

[port]	*integer*	Listening port for the collector.

2.926.4 SysdbExcludesStatus

status	*enum: error | ok | unknown*	Status of the excluded Sysdb path.

2.927 show management api netconf

enabled	*boolean*	Endpoint is enabled.
error	*string*	error which occurred while starting the endpoint.
port	*integer*	The port this protocol's server is listening on.
vrfName	*string*	The VRF this protocol's server is listening in.

2.928 show management api restconf

enabled	*boolean*	Endpoint is enabled.
error	*string*	error which occurred while starting the endpoint.
port	*integer*	The port this protocol's server is listening on.
qosDscp **	*integer*	The QoS DSCP value. **Since: 4.21.0
sslProfile	*string*	SSL profile name.
vrfName	*string*	The VRF this protocol's server is listening in.

2.929 show management api restconf access-list [...]

see 2.53

2.930 show management archive

revision 2

destName	*string*	Nickname of the archival destination.
destPath	*string*	Absolute path of the archival destination.
devForceDisabled	*boolean*	Using the device is not allowed. This is only relevant of the ssd drive.
devMounted	*boolean*	The installed device is mounted.
devName	*string*	Name of the device containing the archival destination.
devPhysical	*boolean*	The device is physical.
devPresent	*boolean*	The device is present on the switch.
exists	*boolean*	The archive symlink and directory tree exist and have the right permissions.
fileArchivalEnabled **	*boolean*	EOS file archival is enabled. **Since: 4.25.1
fsMntPt	*string*	Mountpoint of the filesystem.
fsQuotaMount	*boolean*	The filesystem is mounted with quotas.
fsSize	*integer*	Size of the filesystem in Kibibytes.
fsUsedKiB	*integer*	Filesystem used space in Kibibytes.
fsUsedPct	*integer*	Filesystem used space as a percentage.
quotaKiB	*integer*	Quota limit in Kibibytes.
quotaOn	*boolean*	Quota are enabled for the filesystem.
quotaPct	*integer*	Quota limit as a percentage.
shutdown	*boolean*	True if file archival is disabled.
usedKiB	*integer*	Archive space usage in Kibibytes.
usedPct	*integer*	Archive space usage as a percentage.

2.931 show management cvx

revision 3

clusterStatus	*MgmtCvxClusterStatus* see 2.931.1	Management CVX Cluster status.
[secondaryClusterStatus]	*MgmtCvxClusterStatus* see 2.931.1	Management Secondary CVX Cluster status.

2.931.1 MgmtCvxClusterStatus

controllerStatuses	*table(string) = ControllerStatus* see 2.931.1.1	Controller connection status.
cvxClusterName	*string*	Name of the CVX cluster.
enabled	*boolean*	CVX client state.
heartbeatInterval	*floating point number*	Heartbeat interval in seconds.
heartbeatTimeout	*floating point number*	Heartbeat timeout in seconds.
sourceIntf	*string*	source-interface name.
[sourceIntfAddr]	*IPv4 address*	source-interface address.
[sslProfileStatus]	*SslProfileStatus* see 2.931.1.2	SSL profile status.
[vrfName]	*string*	VRF used to connect to CVX.

2.931.1.1 ControllerStatus

[connectionTime]	*floating point number*	Connection establishment UTC timestamp.
controllerUUID	*string*	Controller UUID.
[duplicatedConnection]	*boolean*	Multiple host entries for the same server.
heartbeatStatus	*HeartbeatStatus* see 2.263.1.1	Heartbeat status of the controller.
hostname **	*string*	Controller hostname. **Since: 4.27.0
[ibConnectionSecured]	*boolean*	Whether in-band connection is SSL secured.
[lastLeaderChangeTs]	*floating point number*	Last leader switchover UTC timestamp.
[leader]	*boolean*	Leader of the CVX cluster.
negotiatedVersion	*integer*	Negotiated version.
[oobConnectionSecured]	*boolean*	Whether out-of-band connection is SSL secured.
oobConnectionStatus	*enum: established | shutdown | connecting | disconnecting | negotiating*	Out-of-band connection status.
versionCompatibility	*enum: compatible | unnegotiated | incompatible | inNegotiation*	Compatibility of switch with controller.

2.931.1.2 SslProfileStatus

[disableReason]	*enum: profileNotExist | dhparamsNotReady | trustedCertNotConfigured | certNotConfigured | profileInvalid | noError*	Reason for disabled SSL profile.
enabled	*boolean*	Whether SSL profile is enabled.
sslProfileName	*string*	SSL profile name.

2.932 show management cvx debug pingpong

see 2.267

2.933 show management cvx mounts

mountStates	*list of MountState* see 2.270.1.1.1.1	State of CVX published mounts.

2.934 show management cvx service

services	*list of MgmtCvxService* see 2.934.1	Status of all services on the switch.

2.934.1 MgmtCvxService

name	*string*	Service name.
status	*ServiceStatus* see 2.272.1.1	Service status.

2.935 show management cvx service NAME

see 2.934.1

2.936 show management dmf controller zerotouch

controllers	*list of IP address*	List of controllers used for ZTN.
enabled	*boolean*	ZTN enabled.
[errorMessage]	*string*	Reason for ZTN handshake failure.
manifestTimestamp	*floating point number*	Time of manifest download in UTC.

state	*enum: manifestReady* | *configDownloaded* | *swiInstallComplete* | *invalidSwi* | *configDownloadInProgress* | *configSaveComplete* | *swiDownloadInProgress* | *manifestDownloadFailed* | *configSaveError* | *configDownloadFailed* | *invalidManifest* | *swiDownloaded* | *init* | *complete* | *rebooting* | *swiDownloadFailed* | *none* | *invalidConfig*	State of ZTN on the switch.

2.937 show management dmf controller zerotouch detail

configDigest	*string*	SHA-256 digest of the downloaded startup-config.
configUrl	*string*	URL used by ZTN to download startup-config.
enabled	*boolean*	ZTN enabled.
imageDigest	*string*	SHA-256 digest of the downloaded image.
imageUrl	*string*	URL used by ZTN to download EOS image.
logMessages	*list of LogMsg* see 2.937.1	Most recent ZTN log messages.
platform	*string*	Platform name sent to the controller for establishing connection.
systemMac	*MAC address*	MAC address of the system sent to the controller for establishing connection.

2.937.1 LogMsg

message	*string*	Log message.
timestamp	*floating point number*	Timestamp of the log.

2.938 show management dmf dynamic configuration

cmds	*table(string) = _Mode or NULL* see 2.938.1	Dictionary of config commands in this mode. The key is a config command. The config command might be: - a simple command and in this case the value is NULL. Example: 'cmds': { 'no ip routing' : NULL } - a command leading to a sub-mode and in this case the value is a new Mode consisting of potential new config commands. Example: 'cmds': { 'interface Ethernet2' : { 'cmds': { } }.
[comments]	*list of string*	Mode comments.

[header]	*list of string*	Header messages. Present only on the Global config mode.

2.938.1 _Mode

Could also be an instance of:
Mode 2.938

2.939 show management http-server

[dscpValue]	*integer*	The QoS DSCP value set in outgoing IP packets.
[fipsEnabled]	*boolean*	Whether FIPS mode is enabled for HTTPS algorithms.
iframeAncestors	*list of string*	Allowed Iframe ancestors.
logLevel	*enum: warn | error | emerg | info | debug | crit | notice | alert | none*	Logging level for the HTTP server.
[sslProfile]	*SslProfile* see 2.921.2	The configuration and status of the SSL profile.
tlsProtocol	*list of string*	Allowed TLS protocol versions.
vrfs	*table(string) = HttpServiceVrf* see 2.939.1	A mapping between the VRF name and http server status.

2.939.1 HttpServiceVrf

httpServer	*HttpProtocolStatus* see 2.921.1	The status of the HTTP server in vrf.
httpsServer	*HttpProtocolStatus* see 2.921.1	The status of the HTTPS server in vrf.
[localHttpServer]	*HttpProtocolStatus* see 2.921.1	The status of localhost HTTP server in vrf.
services	*list of string*	The list of enabled http services in VRF.
[unixSocketServer]	*UnixProtocolStatus* see 2.921.3	The status of the Unix Domain Socket server.

2.940 show management http-server ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.941 show management http-server counters

vrfs	*table(string) = VrfStats* see 2.941.1	A mapping between VRF name and running nginx statistics.

2.941.1 VrfStats

acceptConnections	*integer*	The total number of accepted client connections.
activeConnections	*integer*	The current number of active client connections.
handledConnections	*integer*	The total number of handled connections.
maxConnections	*integer*	The maximum number of simultaneous connections.
readingConnections	*integer*	The current number of connections where nginx is reading the request header.
refusedConnections	*integer*	The total number of refused connections.
requests	*integer*	The total number of client requests.
waitingConnections	*integer*	The current number of connections waiting for a request.
writingConnections	*integer*	The current number of connections where nginx is writing the response back to the client.

2.942 show management ldap

[groups]	*table(string) = ServerGroupModel* see 2.942.3	Mapping of group name to group information.
lastCounterClearTime	*floating point number*	UTC time when the counters were last cleared.
ldapServers	*table(string) = LdapStatsModel* see 2.942.1	Mapping of server IP address or hostname to server statistics.
ldapStatus	*table(string) = LdapStatusModel* see 2.942.2	Mapping of server IP address or hostname to server status.

2.942.1 LdapStatsModel

bindFails	*integer*	Number of failed binds to the LDAP server.
bindRequests	*integer*	Number of bind requests to the LDAP server.
bindSuccesses	*integer*	Number of successful binds to the LDAP server.
bindTimeouts	*integer*	Number of binds timedout to the LDAP server.

2.942.2 LdapStatusModel

fipsStatus	*boolean*	Status of LDAP FIPS mode.

2.942.3 ServerGroupModel

members	*list of ServerInfoModel* see 2.942.3.1	List of LDAP servers in the server group.
serverGroup	*string*	Configured server group.

2.942.3.1 ServerInfoModel

hostname	*string*	IP address or hostname of LDAP server.
port	*integer*	LDAP authorization port number.

2.943 show management package [...]

revision 2

see 2.329

2.944 show management package repository [...]

repositories	*table(string) = RepositoryEntry* see 2.944.1	List of extension repositories.

2.944.1 RepositoryEntry

description	*string*	Repository description.
name	*string*	Repository name.
rType	*enum: yum | pypi | swix | unknown*	Repository type (yum or pypi).
url	*string*	Repository URL.

2.945 show management security

[blockedNetworkProtocols] **	*list of string*	Blocked unencrypted network protocols. **Since: 4.25.0
[cpuModel]	*string*	CPU Model name and manufacturer.
cryptoModule **	*string*	Cryptographic Module name and version. **Since: 4.26.2
entropyServerEnabled **	*boolean*	True if entropy server is serving entropy in the entropy queue. **Since: 4.22.1
hardwareEntropyEnabled	*boolean*	True if the source of entropy is hardware.
[hardwareEntropyQueueSize] **	*integer*	Number of bytes of hardware generated entropy in the queue. **Since: 4.22.0
havegedEntropyEnabled **	*boolean*	True if the source of entropy is haveged. **Since: 4.23.1

`jitterEntropyEnabled` **	*boolean*	True if the source of entropy is jitter. **Since: 4.26.2
`[securityChipVersion]`	*string*	Security chip version. Not present on older platforms.
`[switchChip]`	*string*	Model of the switch chip.

2.946 show management security session shared-secret profile NAME [...]

[currentRxExpiration]	*integer*	The date and time when the current receive secret expires.
[currentRxId]	*integer*	The id of the current secret being used for reception if it can be represented as an integer [deprecated].
[currentTxExpiration]	*integer*	The date and time when the current transmit secret expires.
[currentTxId]	*integer*	The id of the current secret being used for transmition if it can be represented as an integer [deprecated].
[profileName]	*string*	The name associated with the shared secret profile.
[rxId] **	*string*	The id of the current secret being used for reception. **Since: 4.27.0
rxRotation	*list of string*	The order in which receive secrets will be used.
secrets	*list of SecretModel* see 2.946.1	The secrets currently configured in the profile.
[txId] **	*string*	The id of the current secret being used for transmition. **Since: 4.27.0
txRotation	*list of string*	The order in which transmit secrets will be used.

2.946.1 SecretModel

id **	*string*	The id of the secret. **Since: 4.27.0
rxLifetimeEnd	*integer*	The time (UTC) when the secret is no longer valid for receiving data.
rxLifetimeStart	*integer*	The time (UTC) when the secret is valid for receiving data.
secret	*string*	The encrypted secret used for authentication.
secretID	*integer*	The id of the secret when it can be represented as an integer [deprecated].
[timezone] **	*string*	The timezone of the secret lifetime. **Since: 4.29.1
txLifetimeEnd	*integer*	The time (UTC) when the secret is no longer valid for transmitting data.
txLifetimeStart	*integer*	The time (UTC) when the secret is valid for transmitting data.

2.947 show management security session shared-secret profile [...]

profiles	*table(string) = SharedSecretProfile* see 2.946	Shared secret profiles indexed by name.

2.948 show management security ssl certificate [...]

certificates	*table(string) = Certificate* see 2.948.1	Mapping from certificate name to certificate used in SSL/TLS.

2.948.1 Certificate

[extension]	*table(string) = Extension* see 2.948.1.2	Mapping from X.509 version 3 extension name to extension value.
issuer	*DistinguishedName* see 2.948.1.1	Entity who has signed and issued the certificate.
notAfter	*integer*	Timestamp on which the certificate validity period ends.
notBefore	*integer*	Timestamp on which the certificate validity period begins.
publicKey	*PublicKey* see 2.948.1.3	Public key information.
serialNumber	*integer*	Serial number.
subject	*DistinguishedName* see 2.948.1.1	Entity associated with the certificate.
version	*integer*	X.509 version.

2.948.1.1 DistinguishedName

commonName	*string*	Common name.
[country]	*string*	Name of the country.
[email]	*string*	Email address.
[locality]	*string*	Name of the locality.
[organization]	*string*	Name of the organization.
[organizationUnit]	*string*	Name of the organizational unit.
[stateOrProvince]	*string*	Name of the state or province.

2.948.1.2 Extension

critical	*boolean*	Whether the extension is critical.
value	*string*	Value of the X.509 version 3 extension.

2.948.1.3 PublicKey

[ellipticCurve] **	*string*	Elliptic curve used to generate the ECDSA key. This field is not present for other key types. **Since: 4.28.1
encryptionAlgorithm	*enum: ECDSA | RSA*	Encryption algorithm of the key.
modulus	*integer*	Modulus of the key (a value of 0 means no modulus).
[publicExponent]	*integer*	Public exponent of the RSA key. This field is not present for DSA key.
size	*integer*	Size of the key in bits.

2.949 show management security ssl crl [...]

crls	*table(string) = Crl* see 2.949.1	Mapping from CRL name to CRL used in SSL/TLS.

2.949.1 Crl

crlNumber	*integer*	CRL number.
issuer	*DistinguishedName* see 2.948.1.1	Entity who has signed and issued the certificate.
lastUpdate	*integer*	Timestamp on which the CRL validity period begins.
nextUpdate	*integer*	Timestamp on which the CRL validity period ends.
revokedList **	*list of RevokedCertificate* see 2.949.1.1	Serial number and Timestamp of the revoked certificate. **Since: 4.20.0

2.949.1.1 RevokedCertificate

revocationDate	*floating point number*	revocation Date.
serialNumber	*string*	Serial number of the CRL.

2.950 show management security ssl diffie-hellman [...]

[dhparamsLastResetFailed]	*boolean*	Whether last attempt to reset Diffie-Hellman parameters failed.
[dhparamsLastSuccessful-Reset]	*integer*	Last successful Diffie-Hellman parameters reset timestamp.
dhparamsResetInProgress	*boolean*	Whether Diffie-Hellman parameters is being reset.
[diffieHellmanParameters]	*DiffieHellmanParameters* see 2.950.1	Diffie-Hellman parameters.

2.950.1 DiffieHellmanParameters

generator	*integer*	Generator used in Diffie-Hellman key exchange.
prime	*integer*	Prime number used in Diffie-Hellman key exchange.
size	*integer*	Size of the prime number in bits.

2.951 show management security ssl key [...]

publicKeys	*table(string) = PublicKey* see 2.948.1.3	Mapping from key name to public key used in SSL/TLS.

2.952 show management security ssl profile [...]

profileStatus	*table(string) = ProfileStatus* see 2.952.1	Mapping from SSL profile name to status.

2.952.1 ProfileStatus

[certName] **	*string*	Certificate name. **Since: 4.30.0
[chainedCertificates] **	*list of string*	Chained certificates. **Since: 4.30.0
[crls] **	*list of string*	Certificate revocation lists. **Since: 4.30.0
[keyName] **	*string*	Key name. **Since: 4.30.0
[profileError]	*list of ProfileError* see 2.952.1.1	List of SSL profile errors in 'invalid' state.
profileState	*enum: valid* | *invalid* | *updating*	SSL profile state.
[trustedCertificates] **	*list of string*	Trusted certificates. **Since: 4.30.0

2.952.1.1 ProfileError

errorAttr	*enum: trustedCertificate* | *profile* | *chainedCertificate* | *certificate* | *crl* | *key*	SSL profile attribute to which the error applies.
errorAttrValue	*string*	SSL profile attribute value.
errorType	*enum: noProfileData* | *notExist* | *noCABasicConstraintTrust* | *notFqdn* | *noCABasicConstraintChain* | *certTrustChainNotValid* | *noCrlSign* | *fileMultiplePEMs* | *notMatchingCertKey* | *certNotYetValid* | *certExpired* | *noExtendedKeyUsage* | *certChainNotValid* | *hostnameMismatch* | *crlNotSignedByCa* | *noError* | *missingCrlForTrustChain*	Error Type.

2.953 show management security tpm pcr PCR_RANGE [...]

aikCert	*string*	AIK certificate.
quoteData	*string*	Quote data in base64 format.
quoteInfo	*string*	Quote info in base64 format.

registers	*table(numeric string) = string*	A mapping of TPM Platform Configuration Register index to its value (a hexadecimal string).
tpmVersion	*enum: 1.2 | 2.0*	TPM version.

2.954 show management security tpm pcr log [...]

logs	*list of PcrLog* see 2.954.1	TPM Platform Configuration Register extension logs.

2.954.1 PcrLog

context	*list of string*	Additional data/context about this PCR log.
event **	*enum: abootPassword | abootConsoleLogged | abootSwi | eosExtensionInstall | abootSbCert | abootBootConfig | abootImage | eosExtensionUninstall | eosRcEos | unknownEvent | eosStartupConfig | eosExtensionUninstallFailure | eosExtensionInstallFailure | eosKernelParams | abootSbConfig*	Event type of this PCR log. **Since: 4.28.1
extend	*string*	Hexdecimal extended to PCR.
register	*integer*	TPM PCR index.
timestamp	*floating point number*	Timestamp when PCR extended.

2.955 show management ssh ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.956 show management ssh hostkey (KEYALGO | ecdsa) public

hostKey	*string*	The public host key of the switch.

2.957 show management ssh hostkey server cert [...]

certificates	*table(string) = string*	A Mapping between configured host certificate filenames and the raw certificate.

2.958 show management ssh trusted-ca key public [...]

keys	*list of string*	List of trusted CA public keys.

2.959 show management ssh user-keys revoke-list [...]

keys	*list of string*	List of revoked public keys.

2.960 show management telnet ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.961 show management telnet [...]

[globalDefault]	*boolean*	VRF service state is dependent on default telnet state.
maxTelnetSessions	*integer*	Maximum telnet sessions.
maxTelnetSessionsPerHost **	*integer*	Maximum telnet sessions from one host. **Since: 4.24.0
serverState	*enum: disabled | enabled*	State of the telnet session either enabled or disabled.
vrfName	*string*	Configured VRF for telnet session.

2.962 show match-list prefix-ipv4 [...]

matchLists	*table(string) = MatchList* see 2.962.1	A mapping of match list name to prefixes.

2.962.1 MatchList

prefixes	*list of IP address prefix*	List of configured segment prefixes.

2.963 show match-list prefix-ipv6 [...]

see 2.962

2.964 show mdns counters

ignoredKA	*integer*	Ignored Known Answer records.
ignoredNsec	*integer*	Ignored NSEC records.
ignoredProbing	*integer*	Ignored probing queries.
ignoredQuestions	*integer*	Ignored One-shot queries.
rxDsoParseErrors	*integer*	Discarded DSO packets.

rxPamErrors	*integer*	Discarded MDNS packets received on the wrong interface.
rxParseErrors	*integer*	Discarded MDNS packets.
rxTcpPkts	*integer*	Received DSO packets.
rxUdpPkts	*integer*	Received MDNS packets.
suppressedResponseRecords	*integer*	Suppressed MDNS responses.
txDsoGenErrors	*integer*	DSO packet generation errors.
txGenErrors	*integer*	MDNS packet generation errors.
txPamErrors	*integer*	Packet transmission errors.
txTcpPkts	*integer*	Sent DSO packets.
txUdpPkts	*integer*	Sent MDNS packets.

2.965 show mdns links

interfaces	*table(Interface name) = MdnsLinkList* see 2.965.1	mDNS enabled interfaces indexed by interface name.

2.965.1 MdnsLinkList

intfInfos	*list of MdnsLinkEntry* see 2.965.1.1	Interfaces information.

2.965.1.1 MdnsLinkEntry

defaultTag **	*string*	Configured default location tag. **Since: 4.26.2
family	*enum: ipv4 | ipv6 | ipunknown*	Address family.
linkId	*string*	Configured link name.
status	*enum: inactive | active*	Interface status.

2.966 show mdns service name NAME [...]

name	*string*	Service name.
services	*list of MdnsServiceInfo* see 2.966.1	Service information.

2.966.1 MdnsServiceInfo

[hosts]	*table(string) = MdnsServiceHostInfo* see 2.966.1.1	A mapping of Host name to its meta information.
[interface]	*interface name*	Interface the service record is learned from.
[link]	*string*	Name of the link the service record is learned from.
[srvTxts]	*list of Record* see 2.966.1.1.1.1	Service TXT records.

2.966.1.1 MdnsServiceHostInfo

[port]	*integer*	SRV record port.
[priority]	*integer*	SRV record priority.
[txts]	*list of Record* see 2.966.1.1.1.1	TXT records.
[unknownTypes]	*table(numeric string) = MdnsUnknownList* see 2.966.1.1.1	A mapping of unknown types to records.
[v4Addrs]	*list of string*	Information of the A record.
[v6Addrs]	*list of string*	Information of the AAAA record.
[weight]	*integer*	SRV record weight.

2.966.1.1.1 MdnsUnknownList

unknowns	*list of Record* see 2.966.1.1.1.1	Unknown records.

2.966.1.1.1.1 Record

recordBytes	*list of integer*	Raw bytes of records.

2.967 show mdns service rule NAME [...]

queryLinks **	*list of Interface name*	Configured query links. **Since: 4.26.2
responseIntfs **	*list of Interface name*	Configured response interfaces. **Since: 4.26.2
responseLinks **	*list of string*	Configured response links. **Since: 4.26.2

services	*table(string) = MdnsServiceList* see 2.967.1	A mapping of service name to its information.
srvTypes **	*list of string*	Discovering service types. **Since: 4.26.2

2.967.1 MdnsServiceList

srvInfos	*list of MdnsServiceEntry* see 2.967.1.1	mDNS services.

2.967.1.1 MdnsServiceEntry

[interface]	*interface name*	Interface the service record is learned from.
[link]	*string*	Link name the service record is learned from.
locations	*list of string*	Location tags of the service record.

2.968 show mdns service type [...]

serviceTypes	*table(Interface name) = MdnsServiceTypeList* see 2.968.1	Observed service types indexed by interface name.

2.968.1 MdnsServiceTypeList

intfInfos	*list of MdnsServiceTypeEntry* see 2.968.1.1	Interfaces information.

2.968.1.1 MdnsServiceTypeEntry

family	*enum: ipv4 | ipv6 | ipunknown*	Address family.
serviceType	*string*	Service type.
timestamp	*floating point number*	Timestamp of when the service type was last observed.

2.969 show mdns status

dsoEnabled	*boolean*	DSO server is enabled.
dsoRunning	*boolean*	DSO server is running.
mdnsEnabled	*boolean*	mDNS agent is enabled.
mdnsRunning	*boolean*	mDNS agent is running.
tcpClient	*list of MdnsServerGatewayStatus* see 2.969.1	Connections to DSO servers.

tcpServerClient	*list of IpGenericAddrAndPort* see 2.296.1.1.1.2	Connection from DSO clients.

2.969.1 MdnsServerGatewayStatus

client	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Remote gateway.
connectionStatus	*string*	Connection Status.

2.970 show metawatch ptp

revision 2

[ptpClockSummary]	*PtpClockSummary* see 2.970.1	Summary Information for boundary clocks.
ptpIntfSummaries	*table(Interface name) = PtpIntfVlanSummary* see 2.970.2	Collection of summaries of PTP enabled interfaces to display.
[ptpMode]	*enum: ptpOrdinaryMasterClock | ptpOneStepEndToEnd-TransparentClock | ptpDisabled | ptpGeneralized | ptpPeerTo-PeerTransparentClock | ptpBoundaryClock | ptpOneStepBoundaryClock | ptpEndToEndTransparent-Clock*	PTP mode.
[ptpProfile]	*enum: ptpDefaultProfile | ptpG8275_2 | ptpG8275_1*	PTP profile.
[ptpSelectedProfile] **	*enum: ptpDefaultProfile | ptpG8275_2 | ptpG8275_1*	Selected (but not applied) PTP profile. **Since: 4.28.0

2.970.1 PtpClockSummary

clockIdentity	*string*	DefaultDS clock identity.
[currentPtpSystemTime]	*integer*	Current sytem time measured with PTP.
gmClockIdentity	*string*	Grandmaster clock identity. See IEEE 1588-2008 for more information.
[gptpRateRatio]	*floating point number*	Estimated gptp rate ratio.
[lastSyncTime]	*integer*	Last time local clock synced from master.
meanPathDelay	*integer*	CurrentDS mean path delay to master in nanoseconds.
[neighborRateRatio]	*floating point number*	Estimated gptp neighbor rate ratio.
numberOfMasterPorts	*integer*	Number of master ports.
numberOfSlavePorts	*integer*	Number of slave ports.
[offsetFromMaster]	*integer*	CurrentDS offset from master in nanoseconds.
[skew]	*floating point number*	Ratio of master's seconds to local seconds.
[slavePort]	*string*	Interface name of slave port.
[slaveVlanId] **	*integer*	VLAN ID of slave port. **Since: 4.23.1
stepsRemoved	*integer*	CurrentDS steps removed from GM.
[systemClockOffset] **	*integer*	Current offset between system clock and PTP clock in nanoseconds. **Since: 4.26.2

2.970.2 PtpIntfVlanSummary

interface	*interface name*	PTP interface name.
ptpIntfVlanSummaries	*list of PtpIntfSummary* see 2.970.2.1	List of summaries of PTP enabled interface on different VLAN to display. There can be several of them when PTP is configured to run on multiple VLANs on a trunk interface.

2.970.2.1 PtpIntfSummary

[asCapable]	*boolean*	Interface is GPTP capable.
[delayMechanism]	*string*	PTP delay mechanism. See IEEE 1588-2008 for more information.
[lastGptpResidenceTime]	*integer*	Minimum measured GPTP residence time.
[mpassEnabled] **	*boolean*	PTP MPASS operational state. **Since: 4.26.0
[mpassStatus] **	*enum: inactive | active*	PTP MPASS status. **Since: 4.26.0
[neighborRateRatio]	*floating point number*	Estimated gptp neighbor rate ratio.
[peerPathDelay]	*integer*	Peer mean path delay in nanoseconds. See IEEE 1588-2008 for more information.
[portState]	*enum: psMaster | psPassive | psPreMaster | psDisabled | psIsolatedSlave | psSlave | psFaulty | psListening | psInitializing | psUncalibrated*	PTP port state. See IEEE 1588-2008 for more information.
[timeSinceAsCapableLastChanged]	*string*	Time elasped since AS capable state of the port last changed.
transportMode	*enum: ipv4 | ipv6 | layer2*	Type of packet used to send PTP messages.
[vlanId]	*integer*	VLAN on which the PTP trunk interface is configured.

2.971 show mlag [...]

revision 2

[configSanity]	*enum: inconsistent* | *consistent*	Peer configuration consistency check.
[detail]	*Detail* see 2.971.1	More details of MLAG status.
[domainId]	*string*	MLAG domain ID.
[dualPrimaryDetectionState]	*enum: running* | *configured* | *disabled* | *detected*	Dual-primary detection state.
[dualPrimaryMlagRecovery-Delay]	*integer*	How long (in seconds) to delay the recovery of MLAG interfaces after dual primary is resolved.
[dualPrimaryNonMlag-RecoveryDelay]	*integer*	How long (in seconds) to delay the recovery of non-MLAG interfaces after dual primary is resolved.
dualPrimaryPortsErrdisabled	*boolean*	True if ports are errdisabled because dual-primary is detected or recovery delay is active.
[dualPrimaryPorts-ErrdisabledTime]	*floating point number*	Time when the ports were errdisabled due to dual-primary recovery delay.
[heartbeatPeerAddress]	*IPv4 address*	IP address of the other MLAG peer used for UDP heartbeats (if different from peerAddress).
[heartbeatPeerVrf]	*string*	VRF for the heartbeat peer address.
[localInterface]	*interface name*	Layer 3 interface used for communication with MLAG peer.
[localIntfStatus]	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Operational state of the MLAG local interface.
mlagPorts	*table(string) = integer*	Number of MLAGs having each state. Key values are: Disabled, Configured, Inactive, Active-partial, Active-full.
[negStatus]	*enum: invalidPeer* | *connected* | *inNegotiation* | *versionIncompatible* | *connecting* | *hardwareNotReady* | *domainMismatch*	Negotiation status.
[peerAddress]	*IPv4 address*	IP address of the other MLAG peer.
[peerLink]	*interface name*	Layer 2 interface used for communication with MLAG peer.
[peerLinkStatus]	*enum: notPresent* | *lowerLayerDown* | *unknown* | *testing* | *notApplicable* | *down* | *dormant* | *up*	Operational state of the MLAG peer link.
portsErrdisabled	*boolean*	True if ports are disabled because reload delay is active.
[portsErrdisabledTime]	*floating point number*	Time when the ports were errdisabled.

reloadDelay	*integer*	How long (in seconds) the non-peer-link ports are errdisabled after a reboot. -1 means inifinite.
reloadDelayNonMlag	*integer*	How long (in seconds) the non-peer-link, non-MLAG ports are errdisabled after a reboot.
state	*enum: disabled* | *active* | *disabling* | *inactive*	MLAG state.
[systemId]	*MAC address*	MLAG System ID.

2.971.1 Detail

[dualPrimaryAction]	*enum: errdisableAllInterfaces* | *none*	Action to take when dual primary is detected.
[dualPrimaryDetectionDelay]	*integer*	How long (in seconds) to delay the detection of dual-primary after failover.
effectiveHeartbeatInterval	*integer*	Effective time (in milliseconds) between the sending of each MLAG heartbeat.
enabled	*boolean*	True if the MLAG feature is enabled.
failover	*boolean*	True if this MLAG peer is in Failover mode.
failoverCauseList	*list of string*	List of causes for Failover.
failoverInitiated	*boolean*	True if failover is initiated by this MLAG peer due to forceful failover. Examples : Maintenance Mode, Fast Boot Reload.
[fastMacRedirectionEnabled]	*boolean*	True if fast MAC redirection is enabled for faster L2/L3 convergence.
heartbeatInterval	*integer*	Configured time (in milliseconds) between the sending of each MLAG heartbeat.
heartbeatTimeout	*integer*	Time (in milliseconds) of no received MLAG heartbeats before a timeout is declared.
heartbeatTimeoutsSinceReboot	*integer*	Number of MLAG heartbeat timeouts that occurred since the last reload.
lacpStandby	*boolean*	True if LACP standby is used to disable the non-peer-link, MLAG ports during the reload delay window.
[lastFailoverChangeTime]	*floating point number*	Time of the last failover transition.
[lastHeartbeatTimeout]	*floating point number*	Time of the last MLAG heartbeat timeout.
[lastStateChangeTime]	*floating point number*	Time of the last MLAG state transition.
[mlagHwReady]	*boolean*	True if the switch hardware is ready to be configured as an MLAG peer.
[mlagIntfEgressAclInterlockInactiveReason] **	*enum: noPeerConfiguration* | *noInactiveReason* | *noLocalConfiguration* | *mlagNotActive* | *mlagFailover*	Reason why MLAG interface interlock with egress filter is Inactive. **Since: 4.29.0
[mlagIntfEgressAclInterlockState] **	*enum: unconfigured* | *inactive* | *unsupported* | *active*	State of the MLAG interface interlock with egress filter. **Since: 4.29.0

mlagState	*enum: secondary | inactive | disabling | disabled | primary*	Specific MLAG state.
mountChanges	*integer*	Number of MLAG peer mount state transitions.
peerMacAddress	*MAC address*	The other MLAG peer's bridge MAC address.
[peerMacRoutingSupported]	*boolean*	True if the switch hardware can route using peer's bridge MAC address.
peerMlagState	*enum: secondary | primary | unknown*	Peer MLAG state.
[peerMonotonicClockOffset]	*floating point number*	System monotonic time difference (in seconds) between the MLAG peers.
peerPortsErrdisabled	*boolean*	True if the other MLAG peer's ports are disabled because reload delay is active.
peerPrimaryPriority	*integer*	The other MLAG peer's primaryPriority value.
primaryPriority	*integer*	Used to determine which MLAG peer becomes primary when negotiating from scratch.
secondaryFromFailover	*boolean*	True if this MLAG peer's state is secondary because the other peer assumed the primary state due to failover.
stateChanges	*integer*	Number of MLAG state transitions.
udpHeartbeatAlive	*boolean*	True if we received UDP heartbeats from peer within the last timeout period.
udpHeartbeatsReceived	*integer*	Number of MLAG UDP heartbeat received.
udpHeartbeatsSent	*integer*	Number of MLAG UDP heartbeat sent.

2.972 show mlag config-sanity [...]

[detail]	*boolean*	Will list matching peer-local value pairs too.
globalConfiguration	*table(string) = GlobalFeatureParam* see 2.972.1	Global configuration keyed on feature.
interfaceConfiguration	*table(string) = IntfFeatureParam* see 2.972.2	Interface configuration keyed on feature.
mlagActive	*boolean*	Command is enabled if MLAG state is active.
mlagConnected **	*boolean*	Command is enabled if MLAG peer is connected. **Since: 4.26.0

2.972.1 GlobalFeatureParam

globalParameters	*table(string) = ParamValue* see 2.972.1.1	Local and peer values for parameter.

2.972.1.1 ParamValue

[localValue]	*string*	Local value.
[peerValue]	*string*	Peer value.

2.972.2 IntfFeatureParam

interfaceParameters	*table(string) = InterfaceValue* see 2.972.2.1	Local and peer values for interface parameter.

2.972.2.1 InterfaceValue

interface	*table(string) = ParamValue* see 2.972.1.1	Local and peer values for interface.

2.973 show mlag interfaces [...]

interfaces	*table(string) = MlagInterface* see 2.973.1	MLAG interfaces, keyed by MLAG ID.
[mlagIntfEgressAcl-InterlockEnabled] **	*boolean*	Is the MLAG interface egress ACL interlock enabled or not. **Since: 4.29.2

2.973.1 MlagInterface

[detail]	*Detail* see 2.973.1.1	More details of MLAG interface.
[lastTimeoutTime] **	*floating point number*	Time of the last timeout during egress filter interlock. **Since: 4.29.0
[localAclProtocolState] **	*enum: installed | timeout | n/a | pending*	State of local egress ACL protocol. **Since: 4.29.0
[localInterface]	*interface name*	Local interface corresponding to the MLAG.
[localInterfaceDescription]	*string*	Description of the local interface corresponding to the MLAG.
[localInterfaceStatus]	*enum: notPresent | lowerLayerDown | unknown | testing | notApplicable | down | dormant | up*	Operational state of the local MLAG interface.
[members]	*list of Interface name*	MLAG interface members (from both peers).
[peerAclProtocolState] **	*enum: installed | n/a | pending*	State of peer egress ACL protocol. **Since: 4.29.0
[peerInterface]	*interface name*	Peer interface corresponding to the MLAG.
[peerInterfaceStatus]	*enum: notPresent | lowerLayerDown | unknown | testing | notApplicable | down | dormant | up*	Operational state of the peer MLAG interface.
[status]	*enum: configured | inactive | active-partial-local-up | active-partial | disabled | active-full | active-partial-remote-up | n/a*	Status of MLAG.
timeoutCount **	*integer*	Number of timeouts that have occurred during egress filter interlock. **Since: 4.29.0

2.973.1.1 Detail

[changeCount]	*integer*	Number of status transitions.
[lastChangeTime]	*floating point number*	Time of the last status transition.
[localInterfaceEnabled]	*enum: ena | dis*	Enabled state of the local MLAG interface.
[peerInterfaceEnabled]	*enum: ena | dis*	Enabled state of the peer MLAG interface.

2.974 show mlag interfaces interlock filter egress [...]

see 2.973

2.975 show mlag interfaces members [...]

see 2.973

2.976 show mlag issu compatibility URL

revision 2

currentVersion	*string*	The EOS version of the current image.
[detailedErrMessage]	*string*	detailed error message for invalid image.
isValid	*boolean*	True if the new image is valid.
newImagePath	*string*	The path to the new image.
[newVersion]	*string*	The EOS version of the new image.

2.977 show mlag issu warnings

issuWarnings	*list of string*	Warning messages after a reload of an MLAG peer is requested.

2.978 show mlag tunnel [...]

decPkts	*integer*	Decapsulated packets.
encPkts	*integer*	Encapsulated packets.
pktTypeCounters	*table(string) = PktTypeCounters* see 2.978.1	Counters by packet type.
rxPkts	*integer*	Received packets.
txPkts	*integer*	Transmitted packets.

2.978.1 PktTypeCounters

decPkts	*integer*	Decapsulated packets.
encPkts	*integer*	Encapsulated packets.

2.979 show mld membership [...]

interfaces	*table(Interface name) = MldMembershipInterface* see 2.979.1	Map of interfaces for which there are listeners attached.

2.979.1 MldMembershipInterface

groups	*table(IPv6 address) = MldGroup* see 2.979.1.1	Map of groups for which there are local listeners.

2.979.1.1 MldGroup

[compatibilityVersion]	*enum: version1 | version2*	The compatibility version is set to version1 or version2. It is set to version2 by default, but set to version1 whenever an MLDv1 report is received for the group.
[creationTime]	*floating point number*	UTC time at which this group was created.
[expiryTime]	*floating point number*	UTC time at which this group will expire.
[filterMode] **	*enum: exclude | include*	The filter mode may be include or exclude. This mode is based on modes of sources. In case there are no sources, the mode is exclude and it means to exclude no sources. **Since: 4.21.0
[lastReporter]	*IPv6 address*	The last host to send an MLD report for this group.
[sources]	*table(IPv6 address) = MldSource* see 2.979.1.1.1	Map of include or exclude sources for this group.

2.979.1.1.1 MldSource

[creationTime]	*floating point number*	UTC time at which this source was created.
[expiryTime]	*floating point number*	UTC time at which this source will expire.
filterMode	*enum: exclude | include*	The filter mode may be include or exclude. In include mode, packet reception is enabled for this source. In exclude mode, it is not.

2.980 show mld querier [...]

interfaces	*table(Interface name) = MldQuerierInterface* see 2.980.1	Map of querier information for MLD interfaces.

2.980.1 MldQuerierInterface

[generalQueryExpiryTime]	*floating point number*	UTC time at which general query will be sent.
lastListenerQueryCount	*integer*	Configured last listener query count.
lastListenerQueryInterval	*floating point number*	Configured last listener query response interval in seconds.
[otherQuerierExpiryTime]	*floating point number*	UTC time at which the other querier will expire.
querier	*IPv6 address*	Ip Address of the querier on this interface.
queryInterval	*floating point number*	Active query interval in seconds.

queryResponseInterval	*floating point number*	Active maximum query response interval in seconds.
robustness	*integer*	Active robustness on this interface.
startupQueryCount	*integer*	Configured startup query count.
startupQueryInterval	*floating point number*	Effective startup query interval in seconds.
version	*enum: 1 | 2*	The MLD version configured on this interface.

2.981 show mld snooping [...]

[mlagConnectionState] **	*enum: disconnected | connected | unknown | failed | connecting | disconnecting | initializing | n/a*	MLD Snooping agent MLAG connection state. **Since: 4.30.1
[mlagMountState] **	*enum: unknown | mounting | n/a | mounted | unmounted | failed*	MLD Snooping agent MLAG mount state. **Since: 4.30.1
[mldSnoopingState]	*enum: disabled | enabled*	Global state of MLD Snooping.
[robustness]	*integer*	Number of queries sent to age out a port's membership in group.
[vlans]	*table(numeric string) = MldSnoopingVlanInfo* see 2.981.1	A mapping of VLAN's ID to its information.

2.981.1 MldSnoopingVlanInfo

[evpnProxyActive] **	*boolean*	MLD proxying enabled via EVPN. **Since: 4.29.2
floodingTraffic	*boolean*	Flooding traffic to VLAN.
groupsOverrun	*boolean*	There has been an attempt to create more than the maximum number of groups.
maxGroups	*integer*	Maximum number of multicast groups that can join the VLAN.
mldSnoopingState	*enum: disabled | enabled*	State of MLD Snooping.
pruningActive	*boolean*	MLD snooping pruning is active.

2.982 show mld snooping counters [...]

interfaces	*table(Interface name) = MldSnoopingCounters-Interface* see 2.982.1	Map Interface name with its counter details.

2.982.1 MldSnoopingCountersInterface

badChecksumIcmpV6Packets-Received	*integer*	Number of packages received for which ICMP v6 checksum check failed.
badChecksumIpPackets-Received	*integer*	Number of packets received for which IP checksum check failed.
badChecksumPimPackets-Received	*integer*	Number of packets received for which PIM checksum check failed.
badMldQueryReceived	*integer*	Number of invalid MLD querys received.
badMldV2ReportReceived	*integer*	Number of invalid MLD v2 reports received.
[mldQuerySend]	*integer*	Number of MLD querys sent.
[mldReportSend]	*integer*	Number of MLD reports sent.
[mldV1QueryReceived]	*integer*	Number of MLD v1 queries received.
[mldV2QueryReceived]	*integer*	Number of MLD v2 queries received.
[mldV2ReportReceived]	*integer*	Number of MLD v2 reports received.
nonIpPacketsReceived	*integer*	Number of non IP packets received.
[otherIcmpPacketsReceived]	*integer*	Number of other ICMP v6 packets received.
[otherPacketsSent]	*integer*	Number of other packets sent.
[pimPacketsReceived]	*integer*	Number of PIM packets received.
shortPacketsReceived	*integer*	Number of packets received with not enough IP payload.
unknownIpPacketsReceived	*integer*	Number of packets received with unknown IP Protocol.

2.983 show mld snooping groups [...]

see 2.576

2.984 show mld snooping groups count [...]

see 2.577

2.985 show mld snooping groups detail [...]

member	*table(numeric string) = VlanMembershipDetailInfo* see 2.985.1	Map vlanId to per VLAN group membership information.

2.985.1 VlanMembershipDetailInfo

[groupMembership]	*table(IPv4 or IPv6 address) = GroupMembershipDetailInfo* see 2.985.1.1	Map IGMP group to source membership information.
[routerIntf]	*table(Interface name) = MemberInfo* see 2.576.1.1.1	List of router interfaces.
[routerVtep]	*table(string) = MemberInfo* see 2.576.1.1.1	List of router VTEPs.

2.985.1.1 GroupMembershipDetailInfo

sourceMembership	*table(IPv4 or IPv6 address) = SourceMembershipInfo* see 2.985.1.1.1	Map Source address to interested interfaces/VTEPs.

2.985.1.1.1 SourceMembershipInfo

[detail]	*SourceMembershipDetailInfo* see 2.985.1.1.1.1	Additional information about the source.
[excludeIntf]	*table(Interface name) = MemberInfo* see 2.576.1.1.1	Filter mode exclude member ports information.
[excludeVtep]	*table(string) = MemberInfo* see 2.576.1.1.1	Filter mode exclude member VTEPs information.
[includeIntf]	*table(Interface name) = MemberInfo* see 2.576.1.1.1	Filter mode include member ports information.
[includeVtep]	*table(string) = MemberInfo* see 2.576.1.1.1	Filter mode exclude member VTEPs information.

2.985.1.1.1.1 SourceMembershipDetailInfo

creationTime	*floating point number*	The timestamp in UTC when source was created.

2.986 show mld snooping mrouter [...]

see 2.580

2.987 show mld snooping querier [...]

vlans	*table(numeric string) = MldSnoopingVlanQuerier* see 2.987.1	Mapping vlan Id to the vlan querier info.

2.987.1 MldSnoopingVlanQuerier

[adminState] **	*enum: disabled | enabled*	State of MLD querier in the Vlan. **Since: 4.29.0
[listenerTimeout] **	*floating point number*	Time before the querier decides there are no morelisteners for a multicast address in the VLAN. **Since: 4.29.0
mldVersion	*enum: v2 | v1 | unknown*	Version of MLD snooping querier in the vlan.
[operationalState] **	*enum: querierStateInitial | querierStateNonQuerier | querierStateQuerier*	Operational state of querier in the Vlan. **Since: 4.29.0
querierAddress	*IPv6 address*	Address of MLD querier in the VLAN.
querierInterface	*interface name*	Interface where the querier is located.
[queryInterval] **	*floating point number*	Period between MLD Membership Query messages. **Since: 4.29.0
queryResponseInterval	*floating point number*	Effective maximum period a recipient can wait before responding with a membership report in seconds.

2.988 show mld snooping querier counters [...]

rxCounters	*MldSnoopingPacket-Counters* see 2.988.1	RX Counters.
txCounters	*MldSnoopingPacket-Counters* see 2.988.1	TX counters.

2.988.1 MldSnoopingPacketCounters

errorPackets	*integer*	Number of error packets.
otherPackets	*integer*	Number of other packets.
v1Dones	*integer*	Number of v1 dones.
v1GSQueries	*integer*	Number of v1 group specific queries.
v1GeneralQueries	*integer*	Number of v1 general queries.
v1Reports	*integer*	Number of v1 reports.
v2GSQueries	*integer*	Number of v2 group specific queries.
v2GSSQueries	*integer*	Number of v2 group source specific queries.
v2GeneralQueries	*integer*	Number of v2 general queries.
v2Reports	*integer*	Number of v2 reports.

2.989 show mld snooping querier detail [...]

vlans	*table(numeric string) = MldSnoopingVlanQuerier* see 2.987.1	Mapping vlan Id to the vlan querier info.

2.990 show mld snooping querier membership [...]

[vlanMemberships]	*table(numeric string) = VlanMembership* see 2.990.1	Mapping vlan id to the membership info.

2.990.1 VlanMembership

groupMemberships	*table(IPv6 address) = GroupMembership* see 2.990.1.1	Mapping group address to the group info.

2.990.1.1 GroupMembership

filterMode	*enum: filterModeExclude | filterModeInclude*	Filter mode of multicast group.
mldVersion	*enum: v2 | v1 | unknown*	Version of MLD snooping querier.
[sourceMemberships]	*table(IPv6 address) = SourceMembership* see 2.990.1.1.1	Mapping source address to the source info.

2.990.1.1.1 SourceMembership

Could also be an instance of:
GroupMembership 2.990.1.1

mldVersion	*enum: v2 | v1 | unknown*	Version of MLD snooping querier.

2.991 show mld static-group access-list ACL_NAME [...]

[errorMessage]	*string*	Error message for improperly configured access-list.
interfaces	*list of Interface name*	List of the interface names in the VRF using this access-list for static groups.
ruleDetails	*table(numeric string) = RuleInfo* see 2.991.1	Map of sequence number to rule data for this access-list.
totalSGs	*integer*	Total number of valid (S,G) pairs within the access-list across all rules.
totalStarGs	*integer*	Total number of (*,G) pairs within the access-list across all rules.

2.991.1 RuleInfo

[errorMessage]	*string*	Error message for improperly defined rule.
group	*IP address and netmask*	IP Address of the group and mask in this rule.
groupsInRule	*integer*	Number of groups in the specified rule.
source	*IPv6 address*	IP Address of the source in this rule.

2.992 show mld statistics [...]

interfaces	*table(Interface name)* = *MldStatisticsInterface* see 2.992.1	Map of statistics for MLD interfaces.

2.992.1 MldStatisticsInterface

[errorPacketsReceived]	*integer*	Error packets received.
[otherPacketsReceived]	*integer*	Other packets received.
[v1Statistics]	*MldV1StatisticsInterface* see 2.992.1.1	MLD version 1 statistics.
[v2Statistics]	*MldV2StatisticsInterface* see 2.992.1.2	MLD version 2 statistics.

2.992.1.1 MldV1StatisticsInterface

donesReceived	*integer*	MLD v1 multicast listener dones received.
generalQueriesReceived	*integer*	MLD v1 general queries received.
generalQueriesSent	*integer*	MLD v1 general queries sent.
gsQueriesReceived	*integer*	MLD v1 multicast address specific queries received.
gsQueriesSent	*integer*	MLD v1 multicast address specific queries sent.
reportsReceived	*integer*	MLD v1 multicast listener reports received.

2.992.1.2 MldV2StatisticsInterface

generalQueriesReceived	*integer*	MLD v2 general queries received.
generalQueriesSent	*integer*	MLD v2 general queries sent.
gsQueriesReceived	*integer*	MLD v2 multicast address specific queries received.
gsQueriesSent	*integer*	MLD v2 multicast address specific queries sent.
gssQueriesReceived	*integer*	MLD multicast address and source specific queries received.
gssQueriesSent	*integer*	MLD v2 multicast address and source specific queries sent.
reportsReceived	*integer*	MLD v2 multicast listener reports received.

2.993 show mld summary [...]

interfaces	*table(Interface name)* = *MldSummaryInterface* see 2.993.1	Map of summary information for MLD interfaces.

2.993.1 MldSummaryInterface

[ipAddress]	*IPv6 address*	Link Local Ip Address of this interface used in sending queries.
numGroups	*integer*	Number of MLD groups joined on this interface.
querierState	*enum: non-querier* | *querier* | *init*	The querier state may be init, querier, or non-querier.

2.994 show module [...]

modules	*table(string) = Module* see 2.994.1	Mapping between Module's name and its data.
redundancyMode **	*enum: unknown* | *switchover* | *standby* | *disabled* | *active* | *initializing*	informs current redundancy mode. **Since: 4.20.0
redundancyProtocol **	*enum: sso* | *unknownProtocol* | *rpr* | *simplex*	informs current redundancy protocol. **Since: 4.20.0

2.994.1 Module

hardwareRev	*string*	Hardware Revision.
maxMacAddress	*MAC address*	Allocated Module's MAC address range. Inclusive upper bound.
minMacAddress	*MAC address*	Allocated Module's MAC address range. Inclusive lower bound.
modelName	*string*	Module's model string.
numMacAddresses **	*integer*	Number of MAC addresses in range. **Since: 4.22.0
portCount	*integer*	Number of ports.
[powerOffReason] **	*enum: unknown* | *SSU* | *userConfig* | *incompatibleFan* | *tempSensorGroupFailure* | *ejectorOpen* | *overheat* | *insufficientFans*	Reason for being powered off. **Since: 4.28.0
serialNumber	*string*	Serial number.
softwareRev	*string*	Software Revision.
status	*enum: ok* | *upgradingFpga* | *unknown* | *poweringOn* | *standby* | *disabled* | *active* | *poweredOff* | *disabledUntilSystemUpgrade* | *failed*	Module's status. For all Modules: ok-Module operating properly. failed-Module in a failed state. unknown-Unable to determine Module's state. poweringOn-Module is powering on. disabledUntilSystemUpgrade-Module is disabled until system software upgrade. poweredOff-Module is powered off. For supervisor Modules: active-Supervisor is currently active. standby-Supervisor is in standby mode. disabled-Supervisor is disabled.
typeDescription	*string*	Verbose description of the Module's type.

[uptime] **	*floating point number*	UTC timestamp of module power-on completion. **Since: 4.18.0

2.995 show module power [...]

modules	*table(string) = Module* see 2.995.1	A mapping of module name to its power consumption.

2.995.1 Module

inResetPower	*floating point number*	module power consumption when in reset.
moduleModel	*string*	Model of module.
moduleName	*string*	Module slot.
numPresentXcvrs **	*integer*	Number of present transcievers. **Since: 4.27.2
outResetPower	*floating point number*	module power consumption when out of reset.
powerConsumption	*floating point number*	Current power consumption of the module.
powerSources **	*table(string) = PowerSource* see 2.995.1.1	A mapping of power source name to its telemetry. **Since: 4.26.2
presentXcvrs **	*table(string) = string*	A mapping of transciever name to type of inserted module. **Since: 4.27.2
risers **	*table(string) = Riser* see 2.995.1.3	A mapping of riser name to its status and affected interfaces. **Since: 4.30.1
secondaryPowerSources **	*table(string) = SecondaryPowerSource* see 2.995.1.2	A mapping of secondary power source name to its telemetry. **Since: 4.27.2
transceiverPowerSources **	*table(string) = PowerSource* see 2.995.1.1	A mapping of transceiver power source name to its telemetry. **Since: 4.27.1

2.995.1.1 PowerSource

Could also be an instance of:
SecondaryPowerSource 2.995.1.2

current	*floating point number*	Current reading from the power source (amps).
power	*floating point number*	Power reading from the power source (watts).
voltage	*floating point number*	Voltage reading from the power source (volts).

2.995.1.2 SecondaryPowerSource

current	*floating point number*	Current reading from the power source (amps).
firstMediaType	*string*	Type of transciever in the first port.
power	*floating point number*	Power reading from the power source (watts).
secondMediaType	*string*	Type of transciever in the second port.
voltage	*floating point number*	Voltage reading from the power source (volts).

2.995.1.3 Riser

interfaces	*string*	Interfaces on this riser.
powerGood	*boolean*	Power status of the riser (true or false).

2.996 show monitor connectivity [...]

revision 2

vrfs	*table(string) = ConnMonitorVrf* see 2.996.1	A mapping between VRF and the hosts which are probed in it.

2.996.1 ConnMonitorVrf

[clients] **	*table(string) = ConnMonitorClient* see 2.996.1.1	Connectivity monitor clients keyed by name. **Since: 4.29.1
description	*string*	A brief description of the VRF.
hosts	*table(string) = ConnMonitorHost* see 2.996.1.1.1	A mapping between a host name and its probe statistics.

2.996.1.1 ConnMonitorClient

hosts	*table(string) = ConnMonitorHost* see 2.996.1.1.1	A mapping between a host name and its probe statistics.

2.996.1.1.1 ConnMonitorHost

description	*string*	A brief description of the host.
interfaces	*table(string) = Stat* see 2.996.1.1.1.1	A mapping between source interface and the statistics generated using it.
ipAddr	*IP address*	IP address of the host.
pingSize **	*integer*	Size of ping packet in bytes. **Since: 4.27.2
url	*string*	URL of the host.

2.996.1.1.1.1 Stat

hostUnreachable **	*boolean*	Has the sequential failed probe count crossed the threshold. **Since: 4.29.1
httpResponseTime	*floating point number*	HTTP response time in milliseconds to reach the host.
jitter	*floating point number*	Packet delay variance observed while probing the host.
lastResponseAt **	*floating point number*	Timestamp when the last response was received. **Since: 4.29.1
latency	*floating point number*	Network latency in milliseconds to reach the host.
packetLoss	*integer*	Percentage packet loss when reaching the host.
sequentialFailedProbeCount **	*integer*	Number of sequential probes that did not receive any response. **Since: 4.29.1

2.997 show monitor device-health

categories	*table(string) = HealthCategoryModel* see 2.997.1	A mapping between category and category information.
healthMonitoringEnabled	*boolean*	Device health monitoring enabled.

2.997.1 HealthCategoryModel

metrics	*table(string) = HealthMetricModel* see 2.997.1.1	A mapping between metric and metric information.

2.997.1.1 HealthMetricModel

handlers	*list of string*	A mapping between handler and handler information.

2.998 show monitor ethernet reflectors [...]

reflectorInterfaces	*table(string) = ReflectorInterfaceModel* see 2.998.1	Mapping between reflector interface identifiers and their config and status.

2.998.1 ReflectorInterfaceModel

direction	*enum: in | not configured | out*	Direction of the reflector interface.
macAction	*enum: swap | not configured*	MAC action of the reflector interface.
status	*enum: inactive | active*	Status of the reflector interface.
statusReason	*string*	Reason for the status of the reflector interface.

2.999 show monitor hadoop

Could also be an instance of:
ClusterHadoopCounters 2.999.2
ClusterRunningJobs 2.999.2
HadoopCounters 2.999.2

jobs	*table(string) = BasicJobInfo* see 2.999.1	Dictonary of all running jobs on directly attached Nodes. Keyed by composite JobId as shown on Hadoop web interface plus cluster name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

2.999.1 BasicJobInfo

Could also be an instance of:
JobInfo 2.999.2

cluster	*string*	Cluster on which the job is running.
countersPerInterface	*table(Interface name) = CountersPerInterface* see 2.999.1.1	Counters per interface.
jobId	*integer*	Job Id.
jtId	*string*	JobTracker Id.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
mapProgress	*floating point number*	Progress of all map tasks so far.
maps	*integer*	Number of map tasks for this jobs.
name	*string*	Job name.
reduceProgress	*floating point number*	Progress of all reduce tasks so far.
reduces	*integer*	Number of reduce tasks for this jobs.
startTime	*floating point number*	Time when job was started.
taskTrackers	*list of string*	List of Nodes the job is running on.
user	*string*	User for the job.

2.999.1.1 CountersPerInterface

hdfsBytesRead	*integer*	HDFS bytes read so far for this job.
hdfsBytesWritten	*integer*	HDFS bytes written so far for this job.
interface	*interface name*	Interface to which the counters belong.
reduceShuffleBytes	*integer*	Bytes read during reduce shuffle phase.
taskTrackers	*list of string*	List of Nodes on the interface on which counters were collected.

2.999.2 JobInfo

cleanupProgress	*floating point number*	Cleanup progress of the job.
cluster	*string*	Cluster on which the job is running.

countersPerInterface	*table(Interface name) = CountersPerInterface* see 2.999.1.1	Counters per interface.
failureInfo	*string*	Failure information for the job, if any.
jobId	*integer*	Job Id.
jtId	*string*	JobTracker Id.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
mapProgress	*floating point number*	Progress of all map tasks so far.
maps	*integer*	Number of map tasks for this jobs.
name	*string*	Job name.
priority	*enum: veryLow | veryHigh | high | low | none | normal*	Priority of this job.
queueName	*string*	Name of the queue for the job.
reduceProgress	*floating point number*	Progress of all reduce tasks so far.
reduces	*integer*	Number of reduce tasks for this jobs.
runState	*enum: unknown | prep | killed | running | succeeded | failed*	Running state of the job.
schedulingInfo	*string*	Scheduling information for the job.
setupProgress	*floating point number*	Setup progress of the job.
startTime	*floating point number*	Time when job was started.
taskTrackers	*list of string*	List of Nodes the job is running on.
url	*string*	Web URL for the job.
user	*string*	User for the job.
jobs	*table(string) = BasicJobInfo* see 2.999.1	Dictonary of all running jobs on directly attached Nodes. Keyed by composite JobId as shown on Hadoop web interface plus cluster name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
name	*string*	Name of the cluster the jobs are running on.
jobs	*table(string) = BasicJobInfo* see 2.999.1	Dictonary of all running jobs on directly attached Nodes. Keyed by composite JobId as shown on Hadoop web interface plus cluster name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
name	*string*	Name of the cluster the jobs are running on.
jobs	*table(string) = BasicJobInfo* see 2.999.1	Dictonary of all running jobs on directly attached Nodes. Keyed by composite JobId as shown on Hadoop web interface plus cluster name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

see 2.999.2

2.1000 show monitor hadoop cluster CLUSTER_NAME counters

see 2.999.2

2.1001 show monitor hadoop cluster CLUSTER_NAME history

jobHistory	*table(string) = JobHistoryData* see 2.1001.1	List of historical jobs. Keyed by cluster-jobId.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
name	*string*	Name of the cluster the jobs are running on.

2.1001.1 JobHistoryData

cluster	*string*	Cluster on which the job ran.
countersPerInterface	*table(Interface name) = CountersPerInterface* see 2.999.1.1	Counters per interface.
endTime	*floating point number*	End time of the job.
jobId	*integer*	Job Id of the job.
jobName	*string*	Name of the job.
jtId	*string*	JT Id of the JobTracker.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
startTime	*floating point number*	Start time of the job.
user	*string*	User associated with the job.

2.1002 show monitor hadoop cluster CLUSTER_NAME history job JOB

see 2.1001

2.1003 show monitor hadoop cluster CLUSTER_NAME jobs

see 2.999.2

2.1004 show monitor hadoop cluster CLUSTER_NAME jobs JOB

see 2.999.2

2.1005 show monitor hadoop cluster CLUSTER_NAME jobs JOB counters

`jobInfo`	*BasicJobInfo* see 2.999.1	Job information along with the interface counters.
`[lastUpdated]`	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

2.1006 show monitor hadoop cluster CLUSTER_NAME node [...]

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
name	*string*	Name of the cluster Nodes belong to.
taskTrackers	*table(string) = TaskTracker* see 2.1006.1	Dictionary of locally attached Nodes with running task information.

2.1006.1 TaskTracker

cluster	*string*	Name of the cluster to which the Node belongs.
interface	*interface name*	Interface to which the Node is attached.
ipAddress	*IPv4 address*	Ip Address of the Node.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
name	*string*	Node hostname.
runningJobs	*RunningJobs* see 2.999	List of currently running jobs on the Node.
runningTasks	*RunningTasks* see 2.1006.1.1	List of currently running tasks on the Node.
state	*enum: inactive | active*	Node state of active or inactive.
stateDetail	*enum: resolved | noArpEntry | peerArpEntry | unresolvedDnsName | unresolvedArpEntry*	Node state detail.

2.1006.1.1 RunningTasks

interface	*interface name*	Interface to which the Node is attached.
taskReports	*list of TaskReport* see 2.1006.1.1.1	List of running tasks on a Node.
taskTrackerName	*string*	Name of the Node where tasks are running.

2.1006.1.1.1 TaskReport

attemptId	*integer*	Attempt id of the task.
cluster	*string*	Cluster where the task is running.
endTime	*floating point number*	End time of the task.
hdfsBytesRead	*integer*	HDFS bytes read so far for this task.
hdfsBytesWritten	*integer*	HDFS bytes written so far for this task.
interface	*interface name*	Interface on which the traffic was transmitted.
isMap	*boolean*	Whether task is Map or not.
jobId	*integer*	Job id of a job this task belongs to.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
progress	*floating point number*	Progress of the task.

`reduceShuffleBytes`	*integer*	Bytes read during reduce shuffle phase.
`startTime`	*floating point number*	Start time of the task.
`state`	*string*	State of the task.
`status`	*enum: killed* | *running* | *pending* | *complete* | *failed*	Status of the task.
`taskId`	*integer*	Task id of the task.
`taskTrackerName`	*string*	Name of the Node where tasks are running.

2.1007 show monitor hadoop cluster CLUSTER_NAME status

[clusterError]	*ClusterError* see 2.1007.1	Error that was seen during last cluster update, if any.
clusterInfo	*ClusterInfo* see 2.1007.2	Information about the cluster.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
localTaskTrackers	*TaskTrackers* see 2.1007.3	List of Nodes directly attached to the switch.
runningJobs	*RunningJobs* see 2.999	List of currently running jobs on the cluster.

2.1007.1 ClusterError

error	*string*	Error message.
errorSeenAt	*floating point number*	UNIX timestamp in seconds when the error was seen during an update from the JobTracker(MR1)/ResourceManager(MR2).

2.1007.2 ClusterInfo

[allocatedMemory] **	*integer*	Total memory currently allocated across all nodes in MegaBytes(MapReduce2 only). **Since: 4.17.0
clusterConfig	*ClusterConfig* see 2.1007.2.1	Configuration for the cluster.
[jobTrackerHeapSize]	*integer*	Current size of the JobTracker heap in bytes.
[jobTrackerMaxHeapSize]	*integer*	Maximum JobTracker heap size in bytes.
[mapSlots]	*integer*	Number of map slots available in the cluster.
name	*string*	Name of the cluster.
[numActiveTrackers]	*integer*	Number of Nodes currently serving in the cluster.
[numBlacklistedTrackers]	*integer*	Number of Nodes currently in the blacklist of the JobTracker(MR1)/ResourceManager(MR2).
[numDecommissionedNodes]	*integer*	Number of Nodes that were decommissioned from the cluster.
[numMapTasksRunning]	*integer*	Number of map tasks currently running.
[numReduceTasksRunning]	*integer*	Number of reduce tasks currently running.
[operState]	*boolean*	Is cluster data being actively polled from the JobTracker(MR1)/ResourceManager(MR2).
[reduceSlots]	*integer*	Number of reduce slots available in the cluster.
[status]	*enum: running | initializing*	Current status of the JobTracker(MR1)/ResourceManager(MR2).

[totalMemory] **	*integer*	Total memory available across all Nodes in MegaBytes(MapReduce2 only). **Since: 4.17.0
[trackerExpiryInterval]	*floating point number*	Amount of time (in seconds) the JobTracker(MR1)/ResourceManager(MR2) will wait before considering a node dead and its tasks lost.

2.1007.2.1 ClusterConfig

adminStatus	*boolean*	Is cluster administratively enabled or not.
host	*string*	JobTracker hostname.
httpPort	*integer*	HTTP port number for TaskTrackers.
interval	*integer*	Polling interval to poll from the JobTracker using RPC.
[jobHistoryHost] **	*string*	JobHistoryServer hostname. **Since: 4.17.0
[jobHistoryPort] **	*integer*	Port number for the JobHistoryServer. **Since: 4.17.0
[mapReduceVersion] **	*string*	MapReduce Version. **Since: 4.17.0
[resourceManagerHost] **	*string*	ResourceManager hostname. **Since: 4.17.0
[resourceManagerPort] **	*integer*	Port number for the ResourceManager. **Since: 4.17.0
rpcPort	*integer*	RPC port number for the JobTracker.
user	*string*	Username used to connect with the JobTracker.

2.1007.3 TaskTrackers

Could also be an instance of:
ClusterTaskTrackers 2.1006
InterfaceTaskTrackers 2.1007.4

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
taskTrackers	*table(string) = TaskTracker* see 2.1006.1	Dictionary of locally attached Nodes with running task information.

2.1007.4 InterfaceTaskTrackers

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
taskTrackers	*table(string) = TaskTracker* see 2.1006.1	Dictionary of locally attached Nodes with running task information.

2.1008 show monitor hadoop cluster all

clusters	*table(string) = ClusterInfo* see 2.1007.2	List of all the clusters configured.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

2.1009 show monitor hadoop counters

see 2.999.2

2.1010 show monitor hadoop history

Could also be an instance of:
ClusterJobHistory 2.1001

jobHistory	*table(string) = JobHistoryData* see 2.1001.1	List of historical jobs. Keyed by clusterjobId.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

see 2.1001

2.1011 show monitor hadoop node (host HOST) | (interface INTF) counters

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
runningJobs	*table(string) = HadoopCounters* see 2.999.2	Dictionay of HadoopCounters per Node keyed by Node name.

2.1012 show monitor hadoop node (host HOST) | (interface INTF) jobs [...]

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
runningJobs	*table(string) = RunningJobs* see 2.999	Dictionay of running jobs per Node keyed by Node name.

2.1013 show monitor hadoop node (host HOST) | (interface INTF) running-tasks [...]

[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
runningTasksList	*list of RunningTasks* see 2.1006.1.1	List of running tasks per Node.

2.1014 show monitor hadoop node (host HOST) | (interface INTF) running-tasks cluster CLUSTER_NAME job JOB task TASK

see 2.1006.1.1.1

2.1015 show monitor hadoop node all [...]

see 2.1007.3

2.1016 show monitor hadoop node all counters

counters	*table(string) = TaskTrackerCounter* see 2.1016.1	Node counters keyed by Node host name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

2.1016.1 TaskTrackerCounter

bytesIn	*integer*	Total bytes read in by all the jobs running on this Node.
bytesOut	*integer*	Total bytes written in by all the jobs running on this Node.
interface	*interface name*	Interface to which the Node is attached.
ipAddress	*IPv4 address*	Ip Address of the Node.
name	*string*	Node hostname.

2.1017 show monitor hadoop node host HOST status

see 2.1006.1

2.1018 show monitor hadoop node interface INTF status

see 2.1007.4

2.1019 show monitor hadoop status

adminStatus	*boolean*	MapReduce feature is administratively enabled or not.
clusterErrors	*table(string) = ClusterError* see 2.1007.1	Error messages for clusters if there are any. Key is cluster name.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
numClusters	*integer*	Number of clusters configured.
numRunningJobs	*integer*	Number of jobs running locally.
numTaskTrackers	*integer*	Number of local Nodes.
operStatus	*boolean*	Operational status of MapReduce feature.

2.1020 show monitor hadoop traffic burst [...]

bursts	*list of InterfaceBurst* see 2.1020.1	List of bursts per interface.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).

2.1020.1 InterfaceBurst

cluster	*string*	Cluster on which the burst happened.
inBursts	*list of BurstCounter* see 2.1020.1.1	List of top input bursts for a Node during last update interval.
interface	*interface name*	Interface on which the traffic was transmitted.
[lastUpdated]	*floating point number*	UNIX timestamp in seconds of the last update from JobTracker(MR1)/ResourceManager(MR2).
outBursts	*list of BurstCounter* see 2.1020.1.1	List of top output bursts for a Node during last update interval.

2.1020.1.1 BurstCounter

jobId	*integer*	Job Id.
jobName	*string*	Job name.
timestamp	*floating point number*	Time of the burst.
value	*integer*	Burst value.

2.1021 show monitor link qualification generators [...]

enabled **	*boolean*	Link Qualification feature is enabled. **Since: 4.28.2
interfaces	*table(Interface name) = LinkQualification-GeneratorIntfModel* see 2.1021.1	Summary of each generator interface config and the test result, keyed by generator interface.

2.1021.1 LinkQualificationGeneratorIntfModel

duration	*integer*	Duration of the test in seconds.
endTime	*floating point number*	Time (in seconds) when test ended, Unix time.
[errorCode] **	*integer*	Error code to report different failures. **Since: 4.28.2
pktDropped **	*integer*	Number of packets lost. **Since: 4.27.2
pktRcvd	*integer*	Number of packets received.
pktSent	*integer*	Number of packets sent.
pktSize	*integer*	Packet size in bytes.
rate	*integer*	Traffic rate value for the test.
rateUnit **	*enum: rateUnitInvalid | rateUnitbps | rateUnitKbps | rateUnitMbps | rateUnitGbps*	Traffic rate unit for the test. **Since: 4.27.2
rxErrors **	*integer*	Number of receive errors. **Since: 4.27.2
rxOctetCount **	*integer*	Number of octets received. **Since: 4.28.2
startTime	*floating point number*	Time (in seconds) when test was started, Unix time.
state	*enum: cancelled | running | completeWithFailure | complete | none | failed*	Link qualification test state.
txErrors **	*integer*	Number of transmit errors. **Since: 4.27.2
txOctetCount **	*integer*	Number of octets sent. **Since: 4.28.2

2.1022 show monitor link qualification generators profiles [...]

enabled **	*boolean*	Link Qualification feature is enabled. **Since: 4.28.2
profiles	*table(string) = LinkQualificationProfileModel* see 2.1022.1	Summary of each link qualification profile config, keyed by profile name.

2.1022.1 LinkQualificationProfileModel

duration	*integer*	Duration of the link qualification test in seconds.
pktSize	*integer*	Packet size in bytes.
rate	*integer*	Traffic rate value for the link qualification test.
rateUnit **	*enum: rateUnitInvalid | rateUnitbps | rateUnitKbps | rateUnitMbps | rateUnitGbps*	Traffic rate unit for the link qualification test. **Since: 4.27.2

2.1023 show monitor rfc2544 generators profiles [...]

enabled	*boolean*	RFC2544 generator is enabled.
profileNames	*table(string) = Rfc2544ProfileModel* see 2.1023.1	A mapping of profile names to test profiles.
running	*boolean*	RFC2544 generator is running.

2.1023.1 Rfc2544ProfileModel

benchmarkTests	*list of BenchmarkTestModel* see 2.1023.1.1	List of RFC2544 benchmark tests.
destinationMac	*MAC address*	Destination MAC address in test packets.
direction	*enum: flowDirectionUnknown | flowOut | flowIn*	Direction of packets injected by generator.
duration	*floating point number*	Test duration in seconds.
[packetRate]	*PacketRateAndUnitModel* see 2.1023.1.2	Packet rate with unit.
packetSizes	*list of integer*	List of packet sizes.
sourceMac	*MAC address*	Source MAC address in test packets.

2.1023.1.1 BenchmarkTestModel

test	*enum: benchmarkThroughput | benchmarkFrameLossRate*	RFC2544 benchmark test.

2.1023.1.2 PacketRateAndUnitModel

rate	*floating point number*	Packet rate.
unit	*enum: rateUnitGbps | rateUnitMbps | rateUnitKbps | rateUnitbps*	Packet rate unit.

2.1024 show monitor rfc2544 generators test-report [...]

enabled	*boolean*	RFC2544 generator is enabled and running.
interfaces	*table(Interface name) = IntfTestReportModel* see 2.1024.1	A mapping of interfaces to its test reports.

2.1024.1 IntfTestReportModel

testReports	*list of TestReportModel* see 2.1024.1.1	Historic test reports.

2.1024.1.1 TestReportModel

[destinationMac]	*MAC address*	Destination MAC address in test packets.
[direction]	*enum: flowDirectionUnknown | flowOut | flowIn*	Direction of packets injected by generator.
execId	*ExecIdModel* see 2.1024.1.1.1	Unique ID for this test report.
[frameLossRateTest] **	*FrameLossRateTestReportModel* see 2.1024.1.1.2	Frame loss rate test report. **Since: 4.29.0
[profileName]	*string*	RFC2544 inititator profile name.
[sourceMac]	*MAC address*	Source MAC address in test packets.
[throughputTest]	*ThroughputTestReportModel* see 2.1024.1.1.3	Throughput test report.

2.1024.1.1.1 ExecIdModel

count	*integer*	Count.
pid	*integer*	PID.
timestamp	*integer*	UTC timestamp.

2.1024.1.1.2 FrameLossRateTestReportModel

[endTime]	*floating point number*	Timestamp when test was ended.
[rateAndUnit]	*PacketRateAndUnitModel* see 2.1023.1.2	Maximum packet rate.
results	*table(numeric string) = FrameLossRateTestResultModel* see 2.1024.1.1.2.1	A mapping of test packet sizes to frame loss rate test results.
[startTime]	*floating point number*	Timestamp when test was started.
state	*enum: flowStateNone | flowStateRunning | flowStateFailure | flowStateSuccess | flowStateRestart | flowStateAborted*	Overall state of test.

[stateReason]	*enum: eoamAgentRestart* | *throughputFailureWith-ProfileDuration* | *frameLossRateSearch-SpaceExhausted* | *clientEncapNotFound* | *intfDown* | *throughputNotFound* | *noReason* | *hardware-ProgrammingFailed* | *frameLossRateFailureWith-ProfileDuration* | *sourceMacAddrNotFound* | *execRequestStop* | *packetRateExceeded* | *execRequestDeleted*	Reason for test state.

2.1024.1.1.2.1 FrameLossRateTestResultModel

[endTime]	*floating point number*	Timestamp when test was ended.
[startTime]	*floating point number*	Timestamp when test was started.
state	*enum: flowStateNone* | *flowStateRunning* | *flowStateFailure* | *flowStateSuccess* | *flowStateRestart* | *flowStateAborted*	Individual state of test.
[stateReason]	*enum: eoamAgentRestart* | *throughputFailureWith-ProfileDuration* | *frameLossRateSearch-SpaceExhausted* | *clientEncapNotFound* | *intfDown* | *throughputNotFound* | *noReason* | *hardware-ProgrammingFailed* | *frameLossRateFailureWith-ProfileDuration* | *sourceMacAddrNotFound* | *execRequestStop* | *packetRateExceeded* | *execRequestDeleted*	Reason for test state.
trials	*table(numeric string) = FrameLossRateTestTrial-Model* see 2.1024.1.1.2.2	A mapping of test trials to results.

2.1024.1.1.2.2 FrameLossRateTestTrialModel

[duration]	*floating point number*	Trial duration in seconds.
[packetRate]	*PacketRateAndUnitModel* see 2.1023.1.2	Trial packet rate.

[packetsReceived]	*integer*	Number of packets received by the generator.
[packetsSent]	*integer*	Number of packets sent by the generator.
state	*enum: flowStateNone* | *flowStateRunning* | *flowStateFailure* | *flowStateSuccess* | *flowStateRestart* | *flowStateAborted*	State of test trial.
[stateReason]	*enum: eoamAgentRestart* | *throughputFailureWithProfileDuration* | *frameLossRateSearchSpaceExhausted* | *clientEncapNotFound* | *intfDown* | *throughputNotFound* | *noReason* | *hardwareProgrammingFailed* | *frameLossRateFailureWithProfileDuration* | *sourceMacAddrNotFound* | *execRequestStop* | *packetRateExceeded* | *execRequestDeleted*	Reason for test state.

2.1024.1.1.3 ThroughputTestReportModel

[endTime]	*floating point number*	Timestamp when test was ended.
results	*table(numeric string) = ThroughputTestResultModel* see 2.1024.1.1.3.1	A mapping of test packet sizes to throughput test results.
[startTime]	*floating point number*	Timestamp when test was started.
state	*enum: flowStateNone* | *flowStateRunning* | *flowStateFailure* | *flowStateSuccess* | *flowStateRestart* | *flowStateAborted*	Overall state of test.

[stateReason]	*enum: eoamAgentRestart* | *throughputFailureWith-ProfileDuration* | *frameLossRateSearch-SpaceExhausted* | *clientEncapNotFound* | *intfDown* | *throughputNotFound* | *noReason* | *hardware-ProgrammingFailed* | *frameLossRateFailureWith-ProfileDuration* | *sourceMacAddrNotFound* | *execRequestStop* | *packetRateExceeded* | *execRequestDeleted*	Reason for test state.

2.1024.1.1.3.1 ThroughputTestResultModel

[detail]	*ThroughputTestResult-DetailModel* see 2.1024.1.1.3.2	Detailed throughput test trial results.
[packetsSent]	*integer*	Number of packets sent by the generator for this test.
[rateAndUnit]	*PacketRateAndUnitModel* see 2.1023.1.2	Final throughput rate.
state	*enum: flowStateNone* | *flowStateRunning* | *flowStateFailure* | *flowStateSuccess* | *flowStateRestart* | *flowStateAborted*	Individual state of test.
[stateReason]	*enum: eoamAgentRestart* | *throughputFailureWith-ProfileDuration* | *frameLossRateSearch-SpaceExhausted* | *clientEncapNotFound* | *intfDown* | *throughputNotFound* | *noReason* | *hardware-ProgrammingFailed* | *frameLossRateFailureWith-ProfileDuration* | *sourceMacAddrNotFound* | *execRequestStop* | *packetRateExceeded* | *execRequestDeleted*	Reason for test state.

2.1024.1.1.3.2 ThroughputTestResultDetailModel

[endTime]	*floating point number*	Timestamp when test was ended.
[startTime]	*floating point number*	Timestamp when test was started.

trials	*table(numeric string) = ThroughputTestTrialModel* see 2.1024.1.1.3.3	A mapping of test trials to results.

2.1024.1.1.3.3 ThroughputTestTrialModel

[duration]	*floating point number*	Trial duration in seconds.
[packetRate]	*PacketRateAndUnitModel* see 2.1023.1.2	Trial packet rate.
[packetsReceived]	*integer*	Number of packets received by the generator.
[packetsSent]	*integer*	Number of packets sent by the generator.
state	*enum: flowStateNone* | *flowStateRunning* | *flowStateFailure* | *flowStateSuccess* | *flowStateRestart* | *flowStateAborted*	State of test trial.

2.1025 show monitor security awake [...]

nucleuses	*table(string) = MsaNucleus* see 2.1025.1	A mapping of nucleus name to its status.
[status]	*MsaStatus* see 2.1025.2	Monitor security awake status.

2.1025.1 MsaNucleus

destinationAddr	*IP address*	IP address of the nucleus.
destinationPort	*integer*	Port number of the nucleus.
inactiveReasons	*list of MsaNucleusInactiveReason* see 2.1025.1.1	Reasons why the nucleus is not active.
lastEstablished	*floating point number*	Last time that connection was established successfully.
localAddr	*IP address*	Local IP address from which to connect.
localIntf	*interface name*	Local interface from which to connect.
sslProfile	*string*	SSL profile name.
status	*enum: inactive | connected*	Status of the nucleus connection.
vrf	*string*	VRF name.

2.1025.1.1 MsaNucleusInactiveReason

[errorReason]	*string*	TCP or SSL error reason.
inactiveReason	*enum: connectionInProgress | portNotConfigured | tlsVersionsNotConfigured | connectionLost | localAddrNotConfigured | cipherSuiteNotConfigured | topicNotConfigured | vrfNotConfigured | connectionFailed | duplicateDestination | sslProfileNotConfigured | sslProfileNotValid | dnsResolutionInProgress | sslProfileNameNot-Configured | certNotConfigured | hostNotConfigured | trustedCertsNotConfigured*	Nucleus inactive reason.

2.1025.2 MsaStatus

active	*boolean*	Indicates whether the monitor is active or not.
activeIntfs	*list of Interface name*	Interfaces on which monitor is active.
flowTableInactiveTimeout	*floating point number*	Timeout of flows in seconds.
flowTableSize	*integer*	Maximum flow table size.

`inactiveReasons`	*list of MsaInactiveReason* see 2.1025.2.1	Reasons why monitor is not active.
`lowMemoryMode` **	*boolean*	Indicates whether the monitor is running in low memory mode. **Since: 4.30.0
`monitorPointId`	*integer*	Configured monitor point identifier.
`topicName`	*string*	Configured name of topic.

2.1025.2.1 MsaInactiveReason

`inactiveReason`	*enum: monitorSecurity-AwakeDisabled* | *monitorSecurityAwake-Initializing* | *eosExtensionNotInstalled* | *nucleusNotConnected* | *monitorSessionNotActive* | *hwFlowTrackingNot-Running*	Nucleus inactive reason.

2.1026 show monitor security awake counters [...]

[activityRecordQueueFull] **	*integer*	Activity record queue full. **Since: 4.29.0
[flowCounters]	*FlowCounters* see 2.1026.1	Cumulative flow counters.
ipfixCounters	*list of IpfixCounters* see 2.1026.2	A list of per-exporter IPFIX counters.
nucleusCounters	*table(string) = NucleusCounters* see 2.1026.3	A list of per-nucleus counters.
[packetBundleQueueFull] **	*integer*	Packet bundle queue full. **Since: 4.29.0

2.1026.1 FlowCounters

activeFlows	*integer*	Number of active flows.
expiredFlows	*integer*	Cumulative expired flow count.
flows	*integer*	Cumulative created flow count.
flowsRates **	*table(numeric string) = floating point number*	A mapping of time interval to flows created per second. **Since: 4.30.0
ipv4AppCounters	*table(string) = AppCounters* see 2.1026.1.1	A mapping of IPv4 application name to flow counters.
ipv6AppCounters	*table(string) = AppCounters* see 2.1026.1.1	A mapping of IPv6 application name to flow counters.
lastCountersClearedTime	*floating point number*	Last time that the flows counters were-cleared.
packets	*integer*	Cumulative count of packets received.
packetsRates **	*table(numeric string) = floating point number*	A mapping of time interval to packets received per second. **Since: 4.30.0

2.1026.1.1 AppCounters

activeFlows	*integer*	Number of active flows.
expiredFlows	*integer*	Cumulative expired flow count.
flows	*integer*	Cumulative created flow count.

2.1026.2 IpfixCounters

dataRecordsRates **	*table(numeric string) = floating point number*	A mapping of time interval to data records received per second. **Since: 4.30.0
exporter	*IpGenericAddrAndPort* see 2.296.1.1.2	Exporter IP address and port number.
flowRecordQueueFull	*integer*	Flow record queue full.
invalidIpfixMsgs	*integer*	Invalid IPFIX messages received.
lastCountersClearedTime	*floating point number*	Last time that the IPFIX counters were cleared.

messagesRates **	*table(numeric string) = floating point number*	A mapping of time interval to messages received per second. **Since: 4.30.0
observationDomainId	*integer*	Observation domain ID.
rxDataRecords	*integer*	Data records received.
rxMessages	*integer*	Messages received.
rxOptionsDataRecords	*integer*	Options data records received.
rxOptionsTemplateRecords	*integer*	Options template records received.
rxTemplateRecords	*integer*	Template records received.
templateIdErrors	*integer*	Unknown template ID errors.

2.1026.3 NucleusCounters

activityRecordsRates **	*table(numeric string) = floating point number*	A mapping of time interval to activity records transmitted per second. **Since: 4.30.0
connectionErrorCount	*integer*	Number of failed connections.
connectionSuccessCount	*integer*	Number of successful connections.
lastActivityTransmission-Timestamp	*floating point number*	Time when the last activity record was sent.
lastConnectionError-Timestamp	*floating point number*	Time of the last failed connection.
lastConnectionSuccess-Timestamp	*floating point number*	Time of the last successful connection.
lastCountersClearedTime	*floating point number*	Last time that the nucleus counters were cleared.
lastProgressTransmission-Timestamp	*floating point number*	Time when the last progress record was sent.
numActivityRecordsInQueue	*integer*	Number of activity records currently in the queue.
numActivityRecords-Transmitted	*integer*	Number of activity records sent.
numProgressRecordsInQueue	*integer*	Number of progress records currently in the queue.
numProgressRecords-Transmitted	*integer*	Number of progress records sent.
progressRecordsRates **	*table(numeric string) = floating point number*	A mapping of time interval to progress records transmitted per second. **Since: 4.30.0
throughputs **	*table(numeric string) = floating point number*	A mapping of time interval to outgoing activity and progress records throughput in Mbps. **Since: **currentRelease**

2.1027 show monitor security awake dpi memory

[allocated]	*MsaDpiMemoryDetails* see 2.1027.1	Allocated memory details.
[inUse]	*MsaDpiMemoryDetails* see 2.1027.1	In use memory details.

2.1027.1 MsaDpiMemoryDetails

current	*integer*	Current memory use in bytes.
maximum	*integer*	Maximum memory use observed in bytes.
maximumTime	*floating point number*	Time when the maximum memory use was observed.
minimum	*integer*	Minimum memory use observed in bytes.
minimumTime	*floating point number*	Time when the minimum memory use was observed.

2.1028 show monitor security awake dpi memory allocations [...]

contexts	*table(string) = MsaDpiCtxMemoryAllocations* see 2.1028.1	Mapping from context to allocations.

2.1028.1 MsaDpiCtxMemoryAllocations

structures	*table(string) = MsaDpiStructMemoryAllocations* see 2.1028.1.1	Mapping from structure to allocations.

2.1028.1.1 MsaDpiStructMemoryAllocations

maxAlloc	*integer*	Maximum number of allocations observed.
maxBytes	*integer*	Maximum byte allocation observed.
maxTime	*floating point number*	Time when the maximum number of allocations was observed.
minAlloc	*integer*	Minimum number of allocations observed.
minBytes	*integer*	Minimum byte allocation observed.
minTime	*floating point number*	Time when the minimum number of allocations was observed.
numAlloc	*integer*	Current number of allocations.
numBytes	*integer*	Current allocation in bytes.

2.1029 show monitor security awake dpi memory pools

memoryPools	*table(string) = MsaDpiMemoryPoolDetails* see 2.1029.1	Mapping from pool ID to pool details.

2.1029.1 MsaDpiMemoryPoolDetails

allocCount	*integer*	Number of blocks allocated.
allocSize	*integer*	Number of bytes allocated.
blockSize	*integer*	Block size of this memory pool in bytes.
maximumCount	*integer*	Maximum number of blocks in use observed.
maximumSize	*integer*	Maximum number of bytes in use observed.

maximumTime	*floating point number*	Time when the maximum use was observed.
minimumCount	*integer*	Minimum number of blocks in use observed.
minimumSize	*integer*	Minimum number of bytes in use observed.
minimumTime	*floating point number*	Time when the minimum use was observed.
usedCount	*integer*	Number of blocks in use.
usedSize	*integer*	Number of bytes in use.

2.1030 show monitor security awake flow-table [...]

Could also be an instance of:
FlowTableDetail 2.1030.1.1.1

activeFlows	*integer*	Number of active flows.
groups	*table(string) = FlowGroup* see 2.1030.1	A mapping between group name and group.

2.1030.1 FlowGroup

flows	*list of Flow* see 2.1030.1.1	A list of active flows.

2.1030.1.1 Flow

[flowDetail]	*FlowDetail* see 2.1030.1.1.1	Flow detailed information.
higherIpAddress	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Higher IP address.
ipProtocol	*enum: ipProtoExperiment1 | ipProtoRsvp | ipProtoL2tp | ipProtoIpv6NoNxt | ipProtoIpip6 | ipProtoIcmp | ipProtoAh | ipProtoPim | ipProtoIpv6HopByHop | ipProtoTcp | ipProtoUdp | ipProtoIpip | ipProtoIgmp | ipProtoGre | ipProtoIpv6Route | ipProtoSctp | ipProtoIcmpv6 | ipProtoIsis | ipProtoOspf | ipProtoUnknown | ipProtoIpv6Frag | ipProtoIpv6Opts | ipProtoExperiment2 | ipProtoEsp | ipProtoVrrp*	IP protocol.
ipProtocolNumber	*integer*	IP protocol number.
lowerIpAddress	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Lower IP address.
numBytesH2L	*integer*	Number of bytes from higher IP to lower IP.
numBytesL2H	*integer*	Number of bytes from lower IP to higher IP.
numPacketsH2L	*integer*	Number of packets from higher IP to lower IP.
numPacketsL2H	*integer*	Number of packets from lower IP to higher IP.
startTime	*floating point number*	Flow start time.

2.1030.1.1.1 FlowDetail

lastUpdateTime	*floating point number*	Last flow update time.
activeFlows	*integer*	Number of active flows.
groups	*table(string) = FlowGroup* see 2.1030.1	A mapping between group name and group.

2.1031 show monitor server PROTOCOL [...]

[probeMethod]	*enum: statusServerPacket | accessRequestPacket*	Server probe method.
probedServers	*list of ServerProbeStats* see 2.1031.1	List of all probed servers.
protocol	*string*	Protocol of the probed server.

2.1031.1 ServerProbeStats

lastClearTime	*floating point number*	UTC time when the counters were last cleared.
probesSent	*integer*	Number of probe packets sent to the server.
responseAccepted	*integer*	Number of valid responses accepted.
responseDropped	*integer*	Number of responses dropped.
serverInfo	*ServerInfo* see 2.1031.1.1	Probed server information.

2.1031.1.1 ServerInfo

[acctPort]	*integer*	Accounting port number.
alive	*boolean*	Server is alive.
failCount	*integer*	Failed probes since last successful response.
host	*string*	IP address or hostname of the probed server.
lastProbeTime	*floating point number*	UTC time when the server was last probed.
lastResponseTime	*floating point number*	UTC time when receiving the last successful response.
lastStatusChangeTime	*floating point number*	UTC time when the server status was last changed.
port	*integer*	Server port number.
[vrf]	*string*	The VRF over which this server is reachable.

2.1032 show monitor server-failure

enabled	*boolean*	Server-failure feature is enabled or not.
networkServers	*table(string) = integer*	Number of servers in each monitored networkIndexed by network.
[proxyLifetime]	*integer*	Current lifetime.
proxyLifetimeDefault	*integer*	Default proxy lifetime.
proxyService	*boolean*	Is proxy service enabled or not.

2.1033 show monitor server-failure history

fsfEnabled	*boolean*	Is Server-failure feature is enabled or not.
servers	*list of Server* see 2.1033.1	List of servers monitored by Server-failure.

2.1033.1 Server

discoveryTime	*floating point number*	Time at which the server was discovered.
ethAddress	*MAC address*	Server Ethernet Address.
intf	*interface name*	Interface the server is on.
ipAddress	*IPv4 address*	Server IP Address.
isValid	*boolean*	Is the server info valid.
lastFailureTime	*floating point number*	Time at which the server last failed.
lastInactiveTime	*floating point number*	Time at which the server was last made inactive.
lastProxyTime	*floating point number*	Time at which the server was last proxied.
numFailed	*integer*	Number of times the server failed.
numInactive	*integer*	Number of times the server was made inactive.
numProxied	*integer*	Number of times the server was proxied.
state	*string*	Server state.
subnet	*string*	Subnet to which the server belongs.

2.1034 show monitor server-failure servers IP_ADDR

see 2.1033.1

2.1035 show monitor server-failure servers [...]

fsfEnabled	*boolean*	Is Server-failure feature is enabled or not.
servers	*list of Server* see 2.1033.1	List of servers monitored by Server-failure.
summaryStatement	*string*	Summary statement for type of servers in the list.

2.1036 show monitor session [...]

revision 3

multiDestSessions **	*list of string*	List of sessions that have multiple active destinations configured. **Since: 4.23.1
[name]	*string*	The name of the specified session in the CLI command, if it was set.
sessions	*table(string) = MirrorSession* see 2.1036.1	Map from session name to the session details.

2.1036.1 MirrorSession

[aclType]	*enum: mac | ipv6 | ip*	ACL type.
[aclsWithMirrorAction] **	*AclsWithMirrorAction* see 2.1036.1.3	Access lists with mirror action rules. **Since: 4.24.1
[egressAclMirrored-Truncation] **	*integer*	Set to 128 if metadata is enabled for the TX mirroring to GRE. **Since: 4.24.1
greMetadata	*list of string*	Metadata elements in the GRE header.
[grePayloadType]	*enum: payloadTypeNone | payloadTypeFullPacket | payloadTypeInnerPacket*	Payload type of mirrored packets in the session.
[greTimestampingEnabled]	*boolean*	Whether timestamping is enabled for GRE mirror sessions.
greTimestampingSupported	*boolean*	Whether the hardware platform supports GRE timestamping. Can be used by CLI to determine whether or not to report timestamping enabled state.
[greTunnels]	*table(Interface name) = TunnelIntfs* see 2.1036.1.5	Next hop interfaces selected for GRE tunnel destination.
headerRemovalSize **	*integer*	Header removal size in the session. **Since: 4.22.1
hwProgrammingStatus-Supported	*boolean*	Whether platform code supports outputting hardware programming status. Can be used by CLI to indicate whether monitor sessions are programmed in hardware.
[mirrorDeviceName]	*string*	The mirror-device name to use for receiving packets mirrored to cpu.
[mirrorRateLimit]	*integer*	Mirroring rate limit in bps.
[mirrorRateLimitChip]	*enum: per-ingress-chip | per-egress-chip*	Rate limit on multi-chip.
programmedInHw	*boolean*	Whether session is programmed in HW.
[rxSampleRate]	*integer*	Mirroring sampling rate.
sessionAcl	*string*	ACL applied to the session.
[sessionAclDetails] **	*AclDetails* see 2.1036.1.2	Session access-list details. **Since: 4.22.1
[sourceAcls]	*table(string) = Acl* see 2.1036.1.1	The ACL applied to the source interface.

[sourceAclsDetails] **	*table(string) = AclDetails* see 2.1036.1.2	Mapping from source ACL name to its details. **Since: 4.27.0
sourceBothInterfaces	*list of string*	Interfaces mirroring both their ingress and egress traffic.
sourceInactiveInterfaces	*list of Interface* see 2.1036.1.4	Inactive interfaces configured to mirror traffic in this session.
[sourceInactiveVlans] **	*list of Vlan* see 2.1036.1.6	Inactive VLANs configured to mirror traffic in this session. **Since: 4.26.2
sourceInterfaces **	*list of string*	All Interfaces configured as mirroring sources in this session. **Since: 4.22.0
sourceRxInterfaces	*list of string*	Interfaces mirroring their ingress traffic.
[sourceRxVlans] **	*list of integer*	VLANs mirroring their ingress traffic. **Since: 4.26.2
sourceTxInterfaces	*list of string*	Interfaces mirroring their egress traffic.
sourceUnknownInterfaces	*list of string*	Interfaces in an unknown state.
[sourceVlans] **	*list of integer*	All VLANs configured as mirroring sources in this session. **Since: 4.26.2
[targetInterfaces]	*list of Interface* see 2.1036.1.4	Interfaces to which traffic is forwarded.
truncate	*boolean*	Packet truncation enabled in the session.
truncateStatus	*string*	Status and reason for truncation in the session.
[truncationSize]	*integer*	Packet truncation size in the session.
[txMirrorAclSupported] **	*boolean*	Whether tx Mirroring ACLs are supported for mirror sessions. **Since: 4.28.0
[vxlanTunnels] **	*table(Interface name) = TunnelIntfs* see 2.1036.1.5	Next hop interfaces selected for VXLAN tunnel destination. **Since: **currentRelease**

2.1036.1.1 Acl

aclDetails	*boolean*	Display the unsupported TCAM qualifiers.
aclName	*string*	ACL name.
[aclPriority]	*integer*	ACL priority.
[aclType]	*enum: mac | ipv6 | ip*	ACL type.
[aclUnsupportedQualifiers]	*list of string*	Unsupported qualifiers.

2.1036.1.2 AclDetails

[ipAclList]	*IpAclList* see 2.53.1	IPv4 access-lists.
[ipv6AclList]	*Ipv6AclList* see 2.53.2	IPv6 access-lists.
[macAclList]	*MacAclList* see 2.241.1.2.1.1	MAC access-lists.

2.1036.1.3 AclsWithMirrorAction

[ipAcls]	*list of Ipv4AclWithMirrorAction* see 2.1036.1.3.1	IPv4 access lists with mirror action.
[ipv6Acls]	*list of Ipv6AclWithMirrorAction* see 2.1036.1.3.2	IPv6 access lists with mirror action.
[macAcls]	*list of MacAclWithMirrorAction* see 2.1036.1.3.3	MAC access lists with mirror action.

2.1036.1.3.1 Ipv4AclWithMirrorAction

acl	*IpAclList* see 2.53.1	IPv4 access lists.
direction **	*enum: out | in*	Direction of access-list with mirror action. **Since: 4.26.2
name	*string*	ACL name.

2.1036.1.3.2 Ipv6AclWithMirrorAction

acl	*Ipv6AclList* see 2.53.2	IPv6 access lists.
direction **	*enum: out | in*	Direction of access-list with mirror action. **Since: 4.26.2
name	*string*	ACL name.

2.1036.1.3.3 MacAclWithMirrorAction

acl	*MacAclList* see 2.241.1.2.1.1	MAC access lists.
direction **	*enum: out | in*	Direction of access-list with mirror action. **Since: 4.26.2
name	*string*	ACL name.

2.1036.1.4 Interface

[dscp]	*integer*	DSCP of Mirroring Tunnel.
[dstIpGenAddr] **	*IP address*	Mirroring Tunnel destination address. **Since: 4.23.0
[forwardingDrop]	*boolean*	mirror drop packets.
[greHdrProto]	*integer*	protocol in GRE header.
[key] **	*integer*	Key in GRE tunnel. **Since: 4.26.2
name	*interface name*	Interface name.
operState	*enum: active | inactive | unknown*	The interface's current operating status.
operStateReason	*string*	Reason for the current operating status. Only set if the operState is inactive.

[srcIpGenAddr] **	*IP address*	Mirroring Tunnel source address. **Since: 4.23.0
[ttl]	*integer*	TTL of Mirroring Tunnel.
[tunnelType] **	*enum: vxlan | invalid | gre*	Type of mirroring tunnel. **Since: **currentRelease**
[vlan] **	*integer*	VLAN of VXLAN tunnel. **Since: **currentRelease**
[vni] **	*integer*	VNI of VXLAN tunnel. **Since: **currentRelease**
[vrf]	*string*	VRF of Mirroring Tunnel.
[vti] **	*interface name*	VXLAN tunnel interface. **Since: **currentRelease**

2.1036.1.5 TunnelIntfs

tunnelIntfs	*table(Interface name) = MemberIntfs* see 2.1036.1.5.1	tunnel next hop interface.

2.1036.1.5.1 MemberIntfs

memberIntfs	*list of Interface name*	Port-Channel member interfaces.

2.1036.1.6 Vlan

operState	*enum: inactive | unknown*	The interface's current operating status.
[operStateReason]	*string*	Reason the state is inactive. Only set if the operState is inactive.
vlan	*integer*	VLAN ID.

2.1037 show monitor system

currLowMemModeStatus	*boolean*	The system is currently under Memory Exhaustion.
featureEnabled	*boolean*	Memory Exhaustion feature is enabled.
[lastLowMemModeEntered]	*floating point number*	Time of last entrance in Memory Exhaustion.
[lastLowMemModeExited]	*floating point number*	Time of last exit from Memory Exhaustion.
lowMemModeCount	*integer*	Number of times the system has entered Memory Exhaustion.

2.1038 show monitor telemetry inband

deviceId	*string*	Device ID of the switch.
enabled	*boolean*	Inband telemetry is enabled on the switch.
inProgressReason	*string*	Reason for operational status in progress.
intDetectionMethod	*string*	Default Inband telemetry detection method.
intVersion **	*string*	InbandTelemetry version. **Since: 4.29.2
operStatus	*string*	Operational status of Inband telemetry.
[policies]	*table(string) = SamplePolicyModel* see 2.1038.2	Maps sample policy name to its configuration.
probeMarker	*integer*	Configured probe marker for Inband telemetry.
probeProtocol	*integer*	IP header protocol/next-header field for INT packets.
profiles	*InbandTelemetryProfiles* see 2.1038.1	Core, Edge and Vxlan profiles.
udpDstPorts **	*list of integer*	Configured udp destination ports for INT packets. **Since: 4.29.2

2.1038.1 InbandTelemetryProfiles

[coreProfiles]	*table(string) = Profile* see 2.1038.1.1	Maps core profiles to their corresponding configuration.
[edgeProfiles]	*table(string) = Profile* see 2.1038.1.1	Maps edge profiles to their corresponding configuration.
intVersion **	*string*	InbandTelemetry version. **Since: 4.29.2
[vxlanProfiles] **	*table(string) = Profile* see 2.1038.1.1	Maps vxlan profiles to their corresponding configuration. **Since: 4.27.0

2.1038.1.1 Profile

egressCollection	*string*	The collector name.
[egressDrop]	*string*	Whether egress drop is enabled or disabled.
profileErrorReason	*string*	None or the reason for error.
profileStatus	*string*	Whether the port is active or inactive or in error condition.
profileType	*string*	The type of profile.
[sampleRate]	*integer*	Sample rate of the profile.

2.1038.2 SamplePolicyModel

rules	*list of Rule* see 2.1038.2.1	Detailed information of match rules.

2.1038.2.1 Rule

[actions]	*Actions* see 2.1038.2.1.1	Actions for the rule.
matchOption	*string*	Match on packet type.
[matches]	*Matches* see 2.1038.2.1.2	Matches for the rule.
ruleString	*string*	Match rule name.

2.1038.2.1.1 Actions

[count] **	*integer*	(Deprecated - refer to the "inputCounter" attribute) Number of packets hitting the rule applied on input. **Since: 4.23.0
[countNamed] **	*table(string) = CountData* see 2.1038.2.1.1.1	(Deprecated - refer to the "inputNamedCounters" attribute) A mapping of counter name to its input count data. **Since: 4.24.2
[drop]	*boolean*	Drop action.
[goto] **	*GotoAction* see 2.1038.2.1.1.2	Match on the specified rule next. **Since: 4.26.2
[inputCounter] **	*integer*	Number of packets hitting the rule applied on input. **Since: 4.26.0
[inputNamedCounters] **	*table(string) = CountData* see 2.1038.2.1.1.1	A mapping of counter name to its input count data. **Since: 4.26.0
[log] **	*boolean*	Log action. **Since: 4.24.1
[mirrorSession] **	*string*	Set mirror action. **Since: 4.28.2
[outputCounter] **	*integer*	Number of packets hitting the rule applied on output. **Since: 4.26.0
[outputNamedCounters] **	*table(string) = CountData* see 2.1038.2.1.1.1	A mapping of counter name to its output count data. **Since: 4.26.0
[police]	*PoliceRate* see 2.1038.2.1.1.3	Police action.
[redirectNexthop] **	*RedirectNexthop* see 2.1038.2.1.1.4	Action of redirect traffic to next-hop(s). **Since: 4.27.1
[redirectNexthopGroups] **	*list of string*	Action of redirect traffic to next-hop group(s). **Since: 4.26.2
[sample] **	*boolean*	Sample action. **Since: 4.26.0
[sampleAll] **	*boolean*	Sample all action. **Since: 4.26.0
[setDscp] **	*integer*	Set header DSCP value. **Since: 4.23.1
[setHeaderRemove] **	*SetHeaderRemoveModel* see 2.1038.2.1.1.6	Set header remove. **Since: 4.29.1

[setIdentityTag] **	*SetIdentityTagModel* see 2.1038.2.1.1.7	Set Dot1Q identity tag. **Since: 4.27.0
[setMacAddress] **	*SetMacAddressModel* see 2.1038.2.1.1.8	Set MAC address. **Since: 4.28.2
[setTc] **	*integer*	Set forwarding traffic class. **Since: 4.23.1
[setTimestampHeader] **	*boolean*	Set MAC timestamp header. **Since: 4.28.1
[setTtl] **	*integer*	Set header TTL value. **Since: 4.30.0
[setVrf] **	*string*	Set forwarding VRF. **Since: 4.27.0
[setVrfSecondary] **	*SetVrfSecondaryModel* see 2.1038.2.1.1.9	Set secondary forwarding VRF. **Since: 4.28.2
[sflow] **	*boolean*	Sflow action. **Since: 4.29.1
[stripHeaderBytes] **	*string*	Remove 802.1Q outer tag. **Since: 4.27.0
[tapAggRedirect] **	*TapAggRedirect* see 2.1038.2.1.1.10	Redirect traffic action in Tap Aggregation mode. **Since: 4.26.2

2.1038.2.1.1.1 CountData

[byteCount]	*integer*	Number of bytes matched.
[packetCount]	*integer*	Number of packets matched.
[undefined]	*boolean*	Named counter has not been defined.

2.1038.2.1.1.2 GotoAction

[matchName]	*string*	Specified match rule name to go to next.
[nextMatch]	*boolean*	Match on the next match rule.

2.1038.2.1.1.3 PoliceRate

[burstSize] **	*integer*	Lower burst size. **Since: 4.27.2
[burstUnit] **	*string*	Unit of police burst size. **Since: 4.27.2
rate	*integer*	Lower police rate.
unit	*string*	Unit of police rate.

2.1038.2.1.1.4 RedirectNexthop

nexthops	*list of IP address*	List of configured next hops.
[recursive]	*boolean*	Enable recursive next hop resolution.
[resolvedNexthops] **	*table(string) = NexthopInfo* see 2.1038.2.1.1.5	A mapping of ingress VRF name to its active next hops. **Since: 4.27.2

[vrfName]	*string*	Resolve next hop in a VRF.

2.1038.2.1.1.5 NexthopInfo

nexthops	*list of IP address*	List of next hops.

2.1038.2.1.1.6 SetHeaderRemoveModel

preserveEth	*boolean*	Preserve the original Ethernet header, including VLAN tags if any.
size	*integer*	The number of bytes (after the Ethernet header, including any VLAN tags) to remove.

2.1038.2.1.1.7 SetIdentityTagModel

idTag	*integer*	Id-tag value.
innerIdTag	*integer*	Inner id-tag value.

2.1038.2.1.1.8 SetMacAddressModel

destMacAddr	*MAC address*	MAC address for the destination.
srcMacAddr	*MAC address*	MAC address for the source.

2.1038.2.1.1.9 SetVrfSecondaryModel

[useVrfSecondaryResult]	*boolean*	Use secondary VRF route result.
[vrfName]	*string*	Secondary forwarding VRF name.

2.1038.2.1.1.10 TapAggRedirect

groups	*list of string*	List of aggregation groups.
interfaces	*list of string*	List of tool interfaces.

2.1038.2.1.2 Matches

[addressFamily] **	*enum: ipv4 | ipv6*	Address family of encapsulated packet. **Since: 4.26.2
[destExtPrefixSet] **	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Destination prefix field-sets. **Since: 4.26.0
[destLpmPrefixSet] **	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Destination prefix field-sets (longest prefix match). **Since: 4.26.2
[destLpmPrefixes] **	*list of IP address prefix*	Destination prefix (longest prefix match). **Since: 4.26.2

[destPrefixSets] **	*list of string*	(Deprecated - refer to "destExtPrefixSet") Destination prefix field-sets. **Since: 4.23.0
[destPrefixes]	*list of IP address prefix*	Destination prefix.
[dot1qVlanTags] **	*list of Dot1QMatch* see 2.1038.2.1.2.1	802.1Q VLAN tags. **Since: 4.26.2
[dscps]	*list of NumericalRange* see 2.1038.2.1.2.2	DSCP.
[dstMacAddrSet] **	*list of MacAddrFieldSet* see 2.1038.2.1.2.8	Destination MAC field-set. **Since: 4.29.1
[dstMacs] **	*list of MAC address*	Destination MAC addresses. **Since: 4.29.1
[ecns] **	*list of NumericalRange* see 2.1038.2.1.2.2	ECN. **Since: 4.27.1
[encapsulation] **	*Encapsulation* see 2.1038.2.1.2.18	Encapsulation match criteria. **Since: 4.26.2
[ethTypes] **	*list of NumericalRange* see 2.1038.2.1.2.2	(Deprecated - refer to "ethTypesExt") Ethernet type. **Since: 4.29.2
[ethTypesExt] **	*list of HexNumericalRange* see 2.1038.2.1.2.5	Ethernet type. **Since: 4.30.0
[fragment] **	*Fragment* see 2.1038.2.1.2.4	Fragment. **Since: 4.23.1
[innerMatch] **	*MatchesBase* see 2.1038.2.1.2.9	Inner packet match criteria. **Since: 4.26.2
[ipLengths] **	*list of NumericalRange* see 2.1038.2.1.2.2	IP length. **Since: 4.23.2
[locations] **	*list of LocationMatch* see 2.1038.2.1.2.6	Location. **Since: **currentRelease**
[matchIpOptions] **	*boolean*	Match on any IP options. **Since: 4.23.2
[matchL4Protocol]	*string*	Protocol BGP.
[matchTracked] **	*boolean*	Match on any tracked packet. **Since: 4.27.1
[neighborProtocol]	*string*	Neighbor protocol.
[nexthopGroups] **	*list of string*	Next-hop groups. **Since: 4.29.2
[protocols]	*list of Protocol* see 2.1038.2.1.2.11	Protocol.
[srcExtPrefixSet] **	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Source prefix field-sets. **Since: 4.26.0
[srcLpmPrefixSet] **	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Source prefix field-sets (longest prefix match). **Since: 4.26.2
[srcLpmPrefixes] **	*list of IP address prefix*	Source prefix (longest prefix match). **Since: 4.26.2
[srcMacAddrSet] **	*list of MacAddrFieldSet* see 2.1038.2.1.2.8	Source MAC field-set. **Since: 4.29.1
[srcMacs] **	*list of MAC address*	Source MAC addresses. **Since: 4.29.1

[srcPrefixSets] **	*list of string*	Deprecated - refer to "srcExtPrefixSet") Source prefix field-sets. **Since: 4.23.0
[srcPrefixes]	*list of IP address prefix*	Source prefix.
[teidFieldSets] **	*list of TeidFieldSet* see 2.1038.2.1.2.17	TEID field-sets. **Since: 4.28.0
[teids] **	*list of NumericalRange* see 2.1038.2.1.2.2	Tunnel endpoint identifier. **Since: 4.28.0
[ttls]	*list of NumericalRange* see 2.1038.2.1.2.2	TTL.

2.1038.2.1.2.1 Dot1QMatch

[dot1qVlanFieldSets]	*list of VlanFieldSet* see 2.1038.2.1.2.3	Outer VLAN field sets.
[dot1qVlans]	*list of NumericalRange* see 2.1038.2.1.2.2	Outer VLAN tags.
[innerVlans] **	*list of NumericalRange* see 2.1038.2.1.2.2	Inner VLAN tags. **Since: 4.28.2

2.1038.2.1.2.2 NumericalRange

high	*integer*	Maximum value.
low	*integer*	Minimum value.

2.1038.2.1.2.3 VlanFieldSet

[name]	*string*	Name of VLAN field-set.
[numVlans]	*integer*	Number of VLANs in field-set.
[undefined]	*boolean*	VLAN field-set has not been defined.

2.1038.2.1.2.4 Fragment

[fragmentOffsets]	*list of NumericalRange* see 2.1038.2.1.2.2	Fragment offset.
[matchAllFragments]	*boolean*	Match all fragments.

2.1038.2.1.2.5 HexNumericalRange

high	*string*	Maximum value.
low	*string*	Minimum value.

2.1038.2.1.2.6 LocationMatch

[alias]	*string*	Location alias name.
[matches]	*list of ValueMask* see 2.1038.2.1.2.7	Location alias matches.

2.1038.2.1.2.7 ValueMask

[mask]	*string*	Mask for the match.
[value]	*string*	Value to match.

2.1038.2.1.2.8 MacAddrFieldSet

name	*string*	Name of MAC field-set.
[numMacAddresses]	*integer*	Number of MAC addresses in field-set.
[undefined]	*boolean*	MAC field-set has not been defined.

2.1038.2.1.2.9 MatchesBase

Could also be an instance of:
Matches 2.1038.2.1.2

[addressFamily]	*enum: ipv4 | ipv6*	Address family of encapsulated packet.
[destExtPrefixSet]	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Destination prefix field-sets.
[destLpmPrefixSet]	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Destination prefix field-sets (longest prefix match).
[destLpmPrefixes]	*list of IP address prefix*	Destination prefix (longest prefix match).
[destPrefixSets]	*list of string*	(Deprecated - refer to "destExtPrefixSet") Destination prefix field-sets.
[destPrefixes]	*list of IP address prefix*	Destination prefix.
[dot1qVlanTags]	*list of Dot1QMatch* see 2.1038.2.1.2.1	802.1Q VLAN tags.
[dscps]	*list of NumericalRange* see 2.1038.2.1.2.2	DSCP.
[dstMacAddrSet] **	*list of MacAddrFieldSet* see 2.1038.2.1.2.8	Destination MAC field-set. **Since: 4.29.1
[dstMacs] **	*list of MAC address*	Destination MAC addresses. **Since: 4.29.1
[ecns] **	*list of NumericalRange* see 2.1038.2.1.2.2	ECN. **Since: 4.27.1
[encapsulation]	*Encapsulation* see 2.1038.2.1.2.18	Encapsulation match criteria.
[ethTypes] **	*list of NumericalRange* see 2.1038.2.1.2.2	(Deprecated - refer to "ethTypesExt") Ethernet type. **Since: 4.29.2
[ethTypesExt] **	*list of HexNumericalRange* see 2.1038.2.1.2.5	Ethernet type. **Since: 4.30.0
[fragment]	*Fragment* see 2.1038.2.1.2.4	Fragment.
[ipLengths]	*list of NumericalRange* see 2.1038.2.1.2.2	IP length.
[locations] **	*list of LocationMatch* see 2.1038.2.1.2.6	Location. **Since: **currentRelease**
[matchIpOptions]	*boolean*	Match on any IP options.
[matchL4Protocol]	*string*	Protocol BGP.

[matchTracked] **	*boolean*	Match on any tracked packet. **Since: 4.27.1
[neighborProtocol]	*string*	Neighbor protocol.
[nexthopGroups] **	*list of string*	Next-hop groups. **Since: 4.29.2
[protocols]	*list of Protocol* see 2.1038.2.1.2.11	Protocol.
[srcExtPrefixSet]	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Source prefix field-sets.
[srcLpmPrefixSet]	*list of PrefixFieldSet* see 2.1038.2.1.2.10	Source prefix field-sets (longest prefix match).
[srcLpmPrefixes]	*list of IP address prefix*	Source prefix (longest prefix match).
[srcMacAddrSet] **	*list of MacAddrFieldSet* see 2.1038.2.1.2.8	Source MAC field-set. **Since: 4.29.1
[srcMacs] **	*list of MAC address*	Source MAC addresses. **Since: 4.29.1
[srcPrefixSets]	*list of string*	Deprecated - refer to "srcExtPrefixSet") Source prefix field-sets.
[srcPrefixes]	*list of IP address prefix*	Source prefix.
[teidFieldSets] **	*list of TeidFieldSet* see 2.1038.2.1.2.17	TEID field-sets. **Since: 4.28.0
[teids] **	*list of NumericalRange* see 2.1038.2.1.2.2	Tunnel endpoint identifier. **Since: 4.28.0
[ttls]	*list of NumericalRange* see 2.1038.2.1.2.2	TTL.

2.1038.2.1.2.10 PrefixFieldSet

name	*string*	Name of prefix field-set.
[numExceptPrefixes]	*integer*	Number of except prefixes in field-set.
[numPrefixes]	*integer*	Number of accept prefixes in field-set.
[undefined]	*boolean*	Prefix field-set has not been defined.

2.1038.2.1.2.11 Protocol

[icmps] **	*list of IcmpType* see 2.1038.2.1.2.12	ICMP info. **Since: 4.24.2
[ports] **	*list of Port* see 2.1038.2.1.2.13	List of L4 ports that can be ORed together. **Since: 4.25.1
[protocolRange]	*NumericalRange* see 2.1038.2.1.2.2	Protocol.
[tcpFlagGroups] **	*list of TcpFlags* see 2.1038.2.1.2.15	List of TCP flags that can be ORed together. **Since: 4.25.2

2.1038.2.1.2.12 IcmpType

[icmpCodes]	*list of NumericalRange* see 2.1038.2.1.2.2	ICMP code.
[icmpTypes]	*NumericalRange* see 2.1038.2.1.2.2	ICMP type.

2.1038.2.1.2.13 Port

[destExtL4PortFieldSet] **	*list of L4PortFieldSet* see 2.1038.2.1.2.14	Destination l4-port field-set info. **Since: 4.26.0
[destL4PortFieldSets]	*list of string*	(Deprecated - refer to "destExtL4PortFieldSet") Destination l4-port field-sets.
[destPorts]	*list of NumericalRange* see 2.1038.2.1.2.2	Destination port.
[srcExtL4PortFieldSet] **	*list of L4PortFieldSet* see 2.1038.2.1.2.14	Source l4-port field-set info. **Since: 4.26.0
[srcL4PortFieldSets]	*list of string*	(Deprecated - refer to "srcExtL4PortFieldSet") Source l4-port field-sets".
[srcPorts]	*list of NumericalRange* see 2.1038.2.1.2.2	Source port.

2.1038.2.1.2.14 L4PortFieldSet

[name]	*string*	Field-set name.
[numPorts]	*integer*	Number of ports in field-set.
[undefined]	*boolean*	L4 port field-set has not been defined.

2.1038.2.1.2.15 TcpFlags

[tcpFlags]	*list of TcpFlag* see 2.1038.2.1.2.16	List of ANDed TCP flags.

2.1038.2.1.2.16 TcpFlag

name	*string*	TCP flag name.
value	*boolean*	TCP flag bit value.

2.1038.2.1.2.17 TeidFieldSet

[fieldSetType]	*integer*	Type of field-set.
[name]	*string*	Name of TEID field-set.
[numTeids]	*integer*	Number of TEIDs in field-set.
[undefined]	*boolean*	TEID field-set has not been defined.

2.1038.2.1.2.18 Encapsulation

[encapsulationType]	*gtpv1*	Encapsulation type.

2.1039 show monitor telemetry inband profile [...]

see 2.1038.1

2.1040 show monitor telemetry inband profile summary [...]

`[coreIntfList]`	*table(string) = IntfList* see 2.1040.1	Core profiles.
`[edgeIntfList]`	*table(string) = IntfList* see 2.1040.1	Edge profiles.
`[vxlanIntfList]` **	*table(string) = IntfList* see 2.1040.1	Vxlan profiles. **Since: 4.27.0

2.1040.1 IntfList

`[Intfs]`	*list of Interface name*	Interface list.

2.1041 show monitor telemetry influx

`destinationInfluxdb`	*table(string) = DestinationInfluxdb* see 2.1041.1	A mapping between an InfluxDB destination name and an InfluxDB destination context.
`globalTags`	*table(string) = string*	A mapping between a global tag key and a global tag value.
`sourceGroups` **	*table(string) = string*	A mapping between a source group name and its state. **Since: 4.25.2
`sourceSocket`	*table(string) = SourceSocket* see 2.1041.2	A mapping between a socket name and a socket context.
`telegrafRunning`	*boolean*	Telegraf running status.

2.1041.1 DestinationInfluxdb

`databaseName`	*string*	Database name of the InfluxDB destination.
`retentionPolicyName`	*string*	Retention policy name of the InfluxDB destination.
`url`	*string*	URL of the InfluxDB destination.
`vrfName` **	*string*	VRF telegraf will use to transmit data. **Since: 4.25.2

2.1041.2 SourceSocket

`connectionLimit`	*integer*	Maximum number of connections allowed.
`url`	*string*	URL of the socket.

2.1042 show monitor telemetry path characteristics [...]

peers	*table(string) = ItsPeer* see 2.1042.1	A mapping of peer IP address to peer path groups.

2.1042.1 ItsPeer

groups	*table(string) = ItsGroup* see 2.1042.1.1	A mapping of path group name to its paths.
peerName	*string*	Peer name.

2.1042.1.1 ItsGroup

paths	*table(string) = ItsPath* see 2.1042.1.1.1	A mapping of path name to path state.

2.1042.1.1.1 ItsPath

dstIp	*IP address*	Destination IP address of the path.
sessions	*table(numeric string) = ItsSession* see 2.1042.1.1.1.1	A mapping of traffic class to telemetry session.
srcIp	*IP address*	Source IP address of the path.

2.1042.1.1.1.1 ItsSession

[characteristics]	*SessionCharacts* see 2.1042.1.1.1.2	Characteristics of the telemetry session.
[counters]	*SessionCounters* see 2.1042.1.1.1.3	Counters for the telemetry session.

2.1042.1.1.1.2 SessionCharacts

active	*boolean*	The telemetry session state is active.
jitter	*floating point number*	Jitter in milliseconds of the telemetry session.
latency	*floating point number*	One way delay in milliseconds of the telemetry session.
lossRate	*floating point number*	Loss rate in percentage of the telemetry session.
[mtu] **	*integer*	Path MTU. **Since: 4.28.0
seconds	*integer*	Number of seconds the telemetrysession has been in this state.
throughput	*floating point number*	Throughput in Mbps of the telemetry session.

2.1042.1.1.1.3 SessionCounters

inBytes	*integer*	Number of bytes received from the telemetry session.
inPkts	*integer*	Number of packets received from the telemetry session.
inPktsDrop	*integer*	Number of packets dropped after being received from the telemetry session.
outBytes	*integer*	Number of bytes sent out of the telemetry session.
outPkts	*integer*	Number of packets sent out of the telemetry session.
outPktsDrop	*integer*	Number of packets dropped before being sent out of the telemetry session.

2.1043 show monitor telemetry path counters [...]

see 2.1042

2.1044 show monitor telemetry postcard counters

grePktSent	*integer*	GRE packets sent to collector.
multiDstSampleRcvd	*integer*	Muti destination packets sampled by GreenT.
sampleDiscarded	*integer*	Total samples discarded.
sampleRcvd	*integer*	Total packets sampled by GreenT.
sampleSent	*integer*	Total samples sent to collector.

2.1045 show monitor telemetry postcard policy

[collection]	*Collection* see 2.1045.1	Collection attributes.
enabled	*boolean*	Postcard telemetry is enabled.
sampleRate	*integer*	Sampling rate of 1 in 16k, 32k or 64k packets.
[sampleTcpUdpChecksum] **	*SampleTcpUdpChecksum* see 2.1045.2	TCP/UDP checksum and mask. **Since: 4.27.1
vxlanEnabled **	*boolean*	VXLAN marker mode is enabled. **Since: 4.27.1
[vxlanMarkerBit] **	*integer*	VXLAN marker bit. **Since: 4.27.1

2.1045.1 Collection

dstIp	*IP address*	Destination IP address.
srcIp	*IP address*	Source IP address.
version **	*integer*	GRE postcard version. **Since: 4.29.1

2.1045.2 SampleTcpUdpChecksum

mask	*integer*	Mask of TCP/UDP checksum or IP ID.
value	*integer*	TCP/UDP checksum or IP ID value.

2.1046 show monitor telemetry postcard policy profile [...]

profiles	*table(string) = ProfileModel* see 2.1046.1	Maps core profiles to their corresponding configuration.

2.1046.1 ProfileModel

appliedInterfaces	*list of Interface name*	Applied interface list for IPv4 rules.
appliedInterfaces6 **	*list of Interface name*	Applied interface list for IPv6 rules. **Since: 4.27.0
failures **	*table(string) = IntfList* see 2.1046.1.1	Failed interfaces for IPv4 rules, indexed by the failure message. **Since: 4.26.2
failures6 **	*table(string) = IntfList* see 2.1046.1.1	Failed interfaces for IPv6 rules, indexed by the failure message. **Since: 4.27.0
interfaces	*list of Interface name*	Interface list.
[samplePolicy]	*string*	Sample Policy.

2.1046.1.1 IntfList

[interfaces]	*list of Interface name*	Interface list.

2.1047 show monitor telemetry postcard sample policy [...]

policies	*table(string) = SamplePolicyModel* see 2.1047.1	Maps sample policy name to its configuration.

2.1047.1 SamplePolicyModel

rules	*list of Rule* see 2.1038.2.1	Detailed information of match rules.

2.1048 show mpls bgp labeled-unicast bindings [...]

Could also be an instance of:
MplsLdpBindingsModel 2.1048.1.2.1

bindings	*table(string) = MplsLabelBinding* see 2.1048.1	Label bindings.

2.1048.1 MplsLabelBinding

[local]	*MplsLocalLabels* see 2.1048.1.1	Locally defined label bindings.
[remote]	*list of MplsPeerIdAndLabel* see 2.1048.1.2	Remote label bindings.

2.1048.1.1 MplsLocalLabels

[bindingUpdateTimestamp]	*floating point number*	UTC timestamp of when the labels were assigned.
[elc] **	*boolean*	Entropy label capability. **Since: 4.25.1
[filtered] **	*boolean*	FEC is filtered. **Since: 4.17.1
labels	*list of integer*	Local labels.
[sources] **	*BindingSources* see 2.1048.1.1.1	Sources of the binding. **Since: 4.27.0

2.1048.1.1.1 BindingSources

[bgpLu]	*boolean*	BGP labeled-unicast.
[isisSr]	*boolean*	IS-IS SR.
[ldp]	*boolean*	LDP.
[loopbackInterface]	*boolean*	Loopback interface.

2.1048.1.2 MplsPeerIdAndLabel

[bindingUpdateTimestamp]	*floating point number*	UTC timestamp of when the labels were assigned.
[elc] **	*boolean*	Entropy label capability. **Since: 4.25.1
[filtered] **	*boolean*	FEC is filtered. **Since: 4.17.1
labels	*list of integer*	Peer labels.
peer	*MplsPeerRouterId* see 2.1048.1.2.1	Peer ID.

2.1048.1.2.1 MplsPeerRouterId

[hostname] **	*string*	Hostname. **Since: 4.21.3

protocol	*enum: ospf | bgplu | unknown | rsvp | mplste | ldp | isis*	Protocol in which router has this ID.
routerId	*string*	Router ID.
[advertisementsSent]	*table(string) = MplsLdpBindingAdvertisements* see 2.1048.2	Advertisements sent for FEC keyed by FEC.
bindings	*table(string) = MplsLabelBinding* see 2.1048.1	Label bindings.

2.1048.2 MplsLdpBindingAdvertisements

advertisedTo	*list of MplsPeerRouterId* see 2.1048.1.2.1	Peers this FEC binding has been sent to.

2.1049 show mpls config route [...]

revision 3

[isBgpPeerMonitored] **	*boolean*	At the least one via is BGP peer monitored. **Since: 4.28.2
[multiLabelRoutes] **	*table(string) = MplsRoute* see 2.1049.1	Dictionary of multi-label MPLS routes configured, keyed by their top labels, comma-separated, ordered top-most to bottom-most. **Since: 4.22.0
routes	*table(numeric string) = MplsRoute* see 2.1049.1	Dictionary of MPLS routes configured, keyed by their top label.

2.1049.1 MplsRoute

adjacencies	*table(numeric string) = MplsAdjacency* see 2.1049.1.1	Dictionary of adjacencies keyed by metric.
topLabel	*integer*	Incoming MPLS label, set to 1048576 (N/A) if topLabels attribute is present.
[topLabels] **	*string*	Incoming MPLS labels for multi-label routes, comma-separated, ordered top-most to bottom-most. **Since: 4.22.0

2.1049.1.1 MplsAdjacency

[algorithm] **	*string*	The algorithm for the path computation. **Since: 4.26.2
[backupIpLookupVias] **	*list of IpLookupVia* see 2.1049.1.1.5	List of backup IP lookup vias. **Since: 4.26.2
[backupMplsVias] **	*list of MplsVia* see 2.1049.1.1.6	List of backup MPLS vias. **Since: 4.22.1
[evpnEthernetSegmentFilter-Vias]	*list of EvpnEthernet-SegmentFilterVia* see 2.1049.1.1.1	List of EVPN Ethernet Segment Filter vias.
[evpnVlanAwareFloodVias]	*list of EvpnVlanAwareFloodVia* see 2.1049.1.1.2	List of EVPN VLAN Aware Flood vias.
[evpnVlanAwareVias]	*list of EvpnVlanAwareVia* see 2.1049.1.1.3	List of EVPN VLAN Aware vias.
[evpnVlanFloodVias]	*list of EvpnVlanFloodVia* see 2.1049.1.1.4	List of EVPN VLAN Flood vias.
[fecForRoute]	*string*	The FEC for the route.
[fecId] **	*integer*	FEC ID. **Since: 4.24.0
[gribi] **	*RouteGribi* see 2.1049.1.1.9	gRIBI LFIB route info. **Since: 4.23.0
[ipLookupVias]	*list of IpLookupVia* see 2.1049.1.1.5	List of IP lookup vias.

[mplsVias]	*list of MplsVia* see 2.1049.1.1.6	List of MPLS vias.
[multicastIpLookupVias] **	*list of MulticastIpLookupVia* see 2.1049.1.1.7	List of Multicast IP lookup vias. **Since: 4.26.2
[pseudowireVias] **	*list of PwVia* see 2.1049.1.1.8	List of PW vias. **Since: 4.22.1
[source]	*enum: bgpLu* | *isisSrV6Adj* | *staticMcast* | *srp2mp* | *rsvp* | *srtepolicy* | *isisSrToLdp* | *ldp* | *mldp* | *isisSrV4Prefix* | *bgpL3Vpn* | *bgpLuToIsisSr* | *ldpToIsisSr* | *debug* | *isisSrV6Prefix* | *isisSrToBgpLu* | *isisSrV6Area* | *pseudowire* | *gribi* | *isisSrV4Area* | *ldpToBgpLu* | *staticMpls* | *bgpL2Evpn* | *bgpLuToLdp* | *isisSrV4Adj*	LFIB source.
[srTePolicy] **	*RouteSrTePolicy* see 2.691.1.1.1.2	SR-TE policy. **Since: 4.20.5
viaType	*enum: viaTypeMulticastIpLookup* | *viaTypeIpLookup* | *viaTypeMax* | *viaTypeIpLookupBackup* | *viaTypeEvpnVlanFlood* | *viaTypeEvpnEthernet-Segment* | *viaTypeVlan* | *viaTypeNexthopGroup* | *viaTypeEvpnVlanAware* | *viaTypeMplsIp* | *viaType-EvpnVlanAwareFlood* | *viaTypeExternalFec* | *viaTypeMplsUnknown* | *viaTypeMplsPseudowire*	Obsolete because an adjacency can contain mixed via types.
[vlanVias]	*list of VlanVia* see 2.1049.1.1.10	List of VLAN vias.

2.1049.1.1.1 EvpnEthernetSegmentFilterVia

intf	*interface name*	Interface mapping to ESI label.

2.1049.1.1.2 EvpnVlanAwareFloodVia

controlWordEnabled	*boolean*	Control word is enabled.
dot1qTagToVlan	*Dot1qTagToVlan* see 2.1049.1.1.2.1	Dot1Q to VLAN map.

2.1049.1.1.2.1 Dot1qTagToVlan

dot1qTagToVlanId	*table(numeric string) = integer*	Mapping of Dot1Q tags to VLAN IDs.
dot1qTagToVlanKey	*Dot1qTagToVlanKey* see 2.1049.1.1.2.2	Dot1Q to VLAN map key.

2.1049.1.1.2.2 Dot1qTagToVlanKey

index	*integer*	Index of the Dot1Q to VLAN key.
source	*enum: lfibSourceIsisSrV6Adj | lfibSourceIsisSrToLdp | lfibSourceStaticDecapMvpn | lfibSourceUnknown | lfibSourceStaticL3Vpn | lfibSourceOspfSrV4Prefix | lfibSourceLdp | lfibSourceStaticMpls | lfibSourceBgpLu | lfibSourceBgpL2Evpn | lfibSourceOspfSrV4Adj | lfibSourceBgpLuToLdp | lfibSourceMldp | lfibSourceMax | lfibSourceIsisSrV4Prefix | lfibSourceStaticMcast | lfibSourceGribi | lfibSourceLdpToOspfSr | lfibSourceSrP2mp | lfibSourceOspfSrToLdp | lfibSourcePseudowire | lfibSourceIsisSrToBgpLu | lfibSourceRsvp | lfibSourceIsisSrV6Prefix | lfibSourceRsvpAlt | lfibSourceBgpLuToIsisSr | lfibSourceLdpToBgpLu | lfibSourceBgpL3Vpn | lfibSourceIsisSrV4Adj | lfibSourceSrTePolicy | lfibSourceLdpToIsisSr | lfibSourceDebug*	Lfib source.

2.1049.1.1.3 EvpnVlanAwareVia

controlWordEnabled	*boolean*	Control word is enabled.
dot1qTagToVlan	*Dot1qTagToVlan* see 2.1049.1.1.2.1	Dot1Q to VLAN map.

2.1049.1.1.4 EvpnVlanFloodVia

controlWordEnabled	*boolean*	Control word is enabled.
vlanId	*integer*	VLAN ID.

2.1049.1.1.5 IpLookupVia

addressFamily	*enum: ipv4 | ipv6 | autoDecide*	Address family. 'autoDecide' indicates that the lookup decision will be handled dynamically by the platform based on the header following the label.
vrfName	*string*	Name of VRF.

2.1049.1.1.6 MplsVia

[backupVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of backup tunnel vias. **Since: 4.22.0
[bgpPeer] **	*IP address*	Bgp peer IP address. **Since: 4.28.2
[dscpMode] **	*enum: pipe | platformDefault | undefinedTtlMode | uniform*	DSCP mode. **Since: 4.25.1
intf	*interface name*	Egress L3 interface of next hop.
labelAction	*enum: unknown | swap | push | pop | forward*	Label action.
nexthop	*IP address*	Next hop IP address.
[nexthopGroup] **	*string*	Nexthop Group for Via. **Since: 4.26.0
outLabel	*integer*	Out label for the packet.
payloadType	*enum: ipv4 | ipv6 | mpls | undefinedPayload | autoDecide*	Type of payload being carried.
[skipEgressAcl]	*boolean*	Bypass egress ACL for the inner IP packets after MPLS pop.
[srTePolicyId] **	*SrTePolicyId* see 2.151.1.3.1.3	SR-TE policy id. **Since: 4.26.0
[status] **	*enum: down | up*	Resolution status of entry. **Since: 4.22.1
ttlMode	*enum: pipe | platformDefault | undefinedTtlMode | uniform*	TTL mode.
[tunnelDescriptor] **	*TunnelInfo* see 2.151.1.3.1	Tunnel information. **Since: 4.21.0
[vias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel vias. **Since: 4.21.0
[weight] **	*integer*	Weight for the via. **Since: 4.26.0

2.1049.1.1.7 MulticastIpLookupVia

addressFamily	*enum: ipv4 | ipv6 | autoDecide*	Address family. 'autoDecide' indicates that the lookup decision will be handled dynamically by the platform based on the header following the label.
ingressIntf	*string*	Name of the ingress interface.

2.1049.1.1.8 PwVia

connector	*string*	(deprecated; use connectors).
connectors **	*table(string) = FlexibleEncapsulations* see 2.491.1	Local connectors for PW. **Since: 4.26.0
controlWordEnabled	*boolean*	Control word is enabled.
flowLabelEnabled **	*boolean*	Flow label is enabled. **Since: 4.23.2
patch	*string*	Patch associated with PW.
pwType	*enum: pwType5 | pwTypeNone | pwType4*	Pseudowire type.
[vplsInstance] **	*string*	VPLS instance name. **Since: 4.27.0

2.1049.1.1.9 RouteGribi

nhgId	*integer*	gRIBI Nexthop Group ID.

2.1049.1.1.10 VlanVia

controlWordEnabled	*boolean*	Control word is enabled.
vlanId	*integer*	VLAN ID.

2.1050 show mpls hardware ale adj [...]

routes	*table(string) = MplsFecRequestTableEntry* see 2.1050.1	MPLS FEC ID keyed by labels.

2.1050.1 MplsFecRequestTableEntry

installedFecId	*integer*	Installed FEC ID for MPLS route.
programStatus	*string*	Programming status of the requested FEC ID.
requestedAlternateFecIds **	*list of integer*	Requested FEC IDs with alternate configurations for MPLS routes. **Since: 4.28.1
requestedFecId	*integer*	Requested FEC ID for MPLS route.

2.1051 show mpls interface [...]

Could also be an instance of:
MplsInterfaceIngressCounters 2.1051.1

intfs	*table(Interface name) = MplsInterfaceInfo* see 2.1051.1	A mapping between an Interface and MPLS interface info.

2.1051.1 MplsInterfaceInfo

ldpConfigured	*boolean*	Configuration of LDP on interface.
mplsConfigured	*boolean*	Configuration of MPLS on interface.
rsvpConfigured	*boolean*	Configuration of RSVP on interface.
status	*enum: uninitDown | notApplicable | down | adminDown | up*	Operational status of the interface.
interfaces	*table(Interface name) = MplsInterfaceIngress-Counters* see 2.1051.2	MPLS interface ingress counters.
intfs	*table(Interface name) = MplsInterfaceInfo* see 2.1051.1	A mapping between an Interface and MPLS interface info.

2.1051.2 MplsInterfaceIngressCounters

mplsInOctets	*integer*	Number of input MPLS octets.
mplsInPkts	*integer*	Number of input MPLS packets.

2.1052 show mpls interface counters [...]

see 2.1051.1

2.1053 show mpls label ranges

conflictingRanges **	*list of MplsLabelRange* see 2.1053.1	Set of operationally inactive label ranges due to conflict with some other range. **Since: 4.25.1
ranges	*list of MplsLabelRange* see 2.1053.1	Set of operationally active label ranges.

2.1053.1 MplsLabelRange

blockSize	*integer*	The number of labels in the range.
blockStart	*integer*	The first MPLS label in the range.
[protocol]	*string*	The protocol which uses the range, or absent if the range is unused.
rangeType	*enum: dynamic | unassigned | static*	How the range was added.

2.1054 show mpls ldp [...]

`vrfs` **	*table(string) = MplsLdpStatus* see 2.1054.1	MPLS LDP information for all vrfs keyed by vrf name. **Since: 4.17.1

2.1054.1 MplsLdpStatus

`[adjCount]`	*integer*	Number of adjacencies.
`enabled`	*boolean*	LDP enabled/disabled.
`[entropyLabel]` **	*boolean*	Entropy label signaling. **Since: 4.25.0
`[fecCount]`	*integer*	Number of unfiltered FECs.
`[fecFilterPrefixListName]`	*string*	FEC filter prefix list.
`[fecFilteredCount]` **	*integer*	Number of filtered FECs. **Since: 4.18.0
`[grMode]` **	*enum: grModeHelper* | *grModeDisabled* | *grModeSpeaker*	Graceful restart mode. **Since: 4.22.1
`[labelLocalTerminationMode]` **	*enum: implicitNull* | *explicitNull*	Label local termination mode. **Since: 4.23.0
`[linkCount]`	*integer*	Number of links.
`[localGrState]` **	*enum: grStateInactive* | *grStateUnknown* | *grStateActive*	Graceful restart speaker operational state. **Since: 4.22.1
`[localLdpId]`	*MplsLdpId* see 2.1054.1.2	Local LDP ID.
`[localLdpIdIntf]`	*interface name*	Interface used for local LDP ID.
`[mldpBudFecCount]` **	*integer*	Number of bud MLDP FECs. **Since: 4.26.2
`[mldpFecCount]` **	*integer*	Number of MLDP FECs. **Since: 4.26.2
`[mldpLeafFecCount]` **	*integer*	Number of leaf MLDP FECs. **Since: 4.26.2
`[mldpRootFecCount]` **	*integer*	Number of root MLDP FECs. **Since: 4.26.2
`[mldpTransitFecCount]` **	*integer*	Number of transit MLDP FECs. **Since: 4.26.2
`[mldpUnknownFecCount]` **	*integer*	Number of unknown role MLDP FECs. **Since: 4.26.2
`operational`	*boolean*	LDP operational state.
`[parameters]`	*MplsLdpDetail* see 2.1054.1.1	Protocol parameters.
`[peerCount]`	*integer*	Number of peers.
`[reason]`	*string*	Reason for LDP not to be operational.
`[sessionCount]`	*integer*	Number of sessions.
`[transportIp]`	*IP address*	Transport IP.
`[transportIpIntf]`	*interface name*	Interface used for Transport IP.

2.1054.1.1 MplsLdpDetail

advertisedIntfIps	*list of IpGenericAddr* see 2.296.1.1.1.1	Interface addresses advertised to peers.
defaultLdpIntfEnabled **	*boolean*	Default LDP interface state is enabled. **Since: 4.22.0
grMaxNeighborLivenessTime **	*integer*	Graceful restart maximum neighbor liveness timeout in seconds. **Since: 4.22.1
grMaxRecoveryTime **	*integer*	Graceful restart max recovery timeout in seconds. **Since: 4.22.1
grReconnectTime **	*integer*	Graceful restart reconnect timeout in seconds. **Since: 4.22.1
grStateHoldTime **	*integer*	Graceful restart state holding timeout in seconds. **Since: 4.22.1
helloHoldTime	*integer*	Discovery hello hold time in seconds, 65535 is infinite as per RFC 5036.
helloInterval	*integer*	Discovery hello interval in seconds.
initialBackoff	*floating point number*	LDP initial backoff in seconds.
keepAlive	*integer*	Session keepalive time in seconds.
maximumBackoff	*floating point number*	LDP maximum backoff in seconds.
protocolVersion	*integer*	Protcol version.
targetedHelloHoldTime	*integer*	Discovery targeted-hello hold time in seconds, 65535 is infinite as per RFC 5036.
targetedHelloInterval	*integer*	Discovery targeted-hello interval in seconds.

2.1054.1.2 MplsLdpId

Could also be an instance of:
MplsLdpAdjacency 2.1054.1.3

ldpLabelSpace	*integer*	Local label space.
ldpRouterId	*IPv4 address*	LDP Router ID.

2.1054.1.3 MplsLdpAdjacency

[configurationSource]	*list of string*	Configuration source of targeted adjacencies.
[helloDetail]	*MplsLdpAdjacencyDetail* see 2.1054.1.3.1	Detailed hello packet information.
ldpLabelSpace	*integer*	Local label space.
ldpRouterId	*IPv4 address*	LDP Router ID.
[passwordInfo] **	*PasswordInformation* see 2.1054.1.3.2	LDP MD5 password information. **Since: 4.27.2

2.1054.1.3.1 MplsLdpAdjacencyDetail

adjExpireTime **	*floating point number*	UTC time at which discovery entry will expire. **Since: 4.18.1
adjSrcIp	*IP address*	Source IP of adjacency's hello packets.
adjTransportIp	*IP address*	Transport IP of adjacency's hello packets.
holdTime	*integer*	Operational hold time in seconds.
localHoldTime	*integer*	Local proposed hold time in seconds.
peerHoldTime	*integer*	Peer proposed hold time in seconds.

2.1054.1.3.2 PasswordInformation

passwordIndex **	*integer*	MD5 Password index. **Since: 4.27.2
passwordSet	*boolean*	MD5 Password enabled.

2.1055 show mpls ldp access-list [...]

see 2.53

2.1056 show mpls ldp bindings [...]

see 2.1048.1.2.1

2.1057 show mpls ldp bindings mldp [...]

see 2.1048.1.2.1

2.1058 show mpls ldp discovery [...]

vrfs	*table(string) = MplsLdpDiscovery* see 2.1058.1	Discovery information in all vrfs keyed by vrf name.

2.1058.1 MplsLdpDiscovery

discoveryInterfaces	*table(Interface name) = MplsLdpDiscoverySource* see 2.1058.1.1	Link adjacencies discovered on each interface.
discoveryTargets	*table(string) = MplsLdpDiscoverySource* see 2.1058.1.1	Targeted adjacencies discovered from each target.
localLdpId	*MplsLdpId* see 2.1054.1.2	Local LDP ID.
[passwordInfo]	*PasswordInformation* see 2.1054.1.3.2	LDP MD5 password information.

2.1058.1.1 MplsLdpDiscoverySource

`adjacencies`	*list of MplsLdpAdjacency* see 2.1054.1.3	List of adjacencies.
`[helloInterval]`	*integer*	Hello interval in seconds.
`[srcIp]`	*IP address*	Hello source IP address.

2.1059 show mpls ldp neighbor [...]

vrfs	*table(string) = MplsLdpNeighborList* see 2.1059.1	Neighbor sessions in requested vrfs keyed by vrf name.

2.1059.1 MplsLdpNeighborList

neighbors	*list of MplsLdpNeighbor* see 2.1059.1.1	Neighbor sessions in a specific vrf.

2.1059.1.1 MplsLdpNeighbor

boundAddresses	*list of IpGenericAddr* see 2.296.1.1.1.1	Addresses bound to peer.
boundAddressesDup **	*list of IpGenericAddr* see 2.296.1.1.1.1	Addresses bound to peer (duplicate). **Since: 4.24.2
capabilities **	*list of string*	Capabilities advertised by the peer. **Since: 4.20.5
discoveryInterfaces **	*list of Interface name*	LDP Discovery interface sources. **Since: 4.17.1
discoveryTargets **	*list of string*	LDP Discovery target sources. **Since: 4.17.1
endOfLib **	*enum: supported | received | sent | sentAndReceived | notSupported*	End-of-LIB operational state. **Since: 4.24.1
grState **	*enum: notSupported | grStateInactive | grStateUnknown | grStateActive*	Graceful restart operational state. **Since: 4.22.1
localIdent	*MplsLdpId* see 2.1054.1.2	Local LDP ID.
localKeepAliveHoldTime **	*integer*	Local proposed keepalive time in seconds. **Since: 4.21.3
localKeepAliveInterval **	*integer*	Local proposed keepalive transmit interval in seconds. **Since: 4.21.3
msgRx	*integer*	Messages received.
msgTx	*integer*	Messages sent.
peerGrReconnectTime **	*integer*	Peer graceful restart reconnect timeout in seconds. **Since: 4.22.1
peerGrRecoveryTime **	*integer*	Peer graceful restart recovery timeout in seconds. **Since: 4.22.1
peerIdent	*MplsLdpId* see 2.1054.1.2	Peer LDP ID.
peerKeepAliveHoldTime **	*integer*	Peer proposed keepalive time in seconds. **Since: 4.21.3
peerKeepAliveInterval **	*integer*	Peer proposed keepalive transmit interval in seconds. **Since: 4.21.3

`[rxMsgCounters]` **	*LdpMessageCount* see 2.1059.1.1.1	Number of received messages, categorized by type. **Since: 4.18.1
`[rxNotifCounters]` **	*LdpNotificationCount* see 2.1059.1.1.2	Number of received notifications, categorized by their status code. **Since: 4.18.1
`sessionExpireTime` **	*floating point number*	Session expiry in UTC, if no further session messages received. **Since: 4.18.1
`sessionKeepAliveHoldTime` **	*integer*	Negotiated session keepalive time in seconds. **Since: 4.21.3
`sessionKeepAliveInterval` **	*integer*	Negotiated session keepalive transmit interval in seconds. **Since: 4.21.3
`[sessionStateLog]` **	*list of LdpSessionStateLogEntryModel* see 2.1059.1.1.3	The last few session states and their information. **Since: 4.18.0
`sessionUpTimestamp`	*floating point number*	UTC timestamp of when the session went up.
`state`	*enum: stateInitialized* | *stateOpenRecv* | *stateOperational* | *stateExpired* | *stateUnknown* | *stateNonExist* | *stateOpenSent*	Ldp session state.

[stateCode] **	*enum: keepAliveExpired* | *badLdpId* | *unknown-MessageTypeIgnore* | *sessionRoleChanged* | *rejectParamLabelRange* | *noLabelResources* | *unknownMessageType* | *endOfLib* | *noAdjacency* | *authKeyChange* | *idChange* | *loopDetected* | *establishingTcpConnection* | *missingMessageParams* | *shutdown* | *tcpConnectionClosed* | *unknownFec* | *pwStatus* | *grConfigChange* | *badPduLength* | *badMessageLength* | *labelResourcesAvailable* | *targetAdjRemoved* | *unknownTlv* | *malformedTlv* | *sessionRejectBadKeep-AliveTime* | *rejectParamAdvMode* | *eolConfigChange* | *labelRequestAborted* | *neighborCleared* | *mLdpChange* | *holdTimerExpired* | *rejectMaxPduLength* | *illegalCBit* | *badTlvLength* | *wrongCBit* | *sessionInitFailed* | *badVersion* | *internalError* | *rejectNoHello* | *localTransportAddrChange* | *unsupportedAddressFamily* | *peerAddrChange* | *noRoute* | *linkAdjRemoved* | *success*	In states non-exist or expired, the code used for the last state transition. **Since: 4.18.0
[stateCodeDescription] **	*string*	In states non-exist or expired, the reason describing the last state transition. **Since: 4.18.0
stream	*enum: unknownAdvDspl* | *onDemand* | *unSolicited*	Ldp advertisement discipline.
[tcpLocalIp]	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Local IP address.
[tcpPeerIp]	*IpGenericAddrAndPort* see 2.296.1.1.1.2	Peer IP address.
[txMsgCounters] **	*LdpMessageCount* see 2.1059.1.1.1	Number of transmitted messages, categorized by type. **Since: 4.18.1

[txNotifCounters] **	*LdpNotificationCount* see 2.1059.1.1.2	Number of received notifications, categorized by their status code. **Since: 4.18.1

2.1059.1.1.1 LdpMessageCount

ldpAddress	*integer*	Number of LDP Address messages.
ldpAddressWithdraw	*integer*	Number of LDP AddressWithdraw messages.
ldpInit	*integer*	Number of LDP Init messages.
ldpKeepAlive	*integer*	Number of LDP KeepAlive messages.
ldpLabelAbortReq	*integer*	Number of LDP LabelAbortReq messages.
ldpLabelMapping	*integer*	Number of LDP LabelMapping messages.
ldpLabelRelease	*integer*	Number of LDP LabelRelease messages.
ldpLabelRequest	*integer*	Number of LDP LabelRequest messages.
ldpLabelWithdraw	*integer*	Number of LDP LabelWithdraw messages.
ldpNotification	*integer*	Number of LDP Notification messages.

2.1059.1.1.2 LdpNotificationCount

authKeyChange **	*integer*	Number of notification messages with status code 'Authentication Key Change'. **Since: 4.27.1
badLdpId	*integer*	Number of notification messages with status code 'Bad LDP Identifier'.
badMessageLength	*integer*	Number of notification messages with status code 'Bad Message Length'.
badPduLength	*integer*	Number of notification messages with status code 'Bad PDU Length'.
badTlvLength	*integer*	Number of notification messages with status code 'Bad TLV Length'.
badVersion	*integer*	Number of notification messages with status code 'Bad Protocol Version'.
endOfLib **	*integer*	Number of notification messages with status code 'End-of-LIB'. **Since: 4.24.0
eolConfigChange **	*integer*	Number of notification messages with status code 'End-of-LIB Config Change'. **Since: 4.24.1
establishingTcpConnection **	*integer*	Number of notification messages with status code 'Establishing Tcp Connection'. **Since: 4.21.3
grConfigChange **	*integer*	Number of notification messages with status code 'Gr Destructive Config Change'. **Since: 4.22.1
holdTimerExpired	*integer*	Number of notification messages with status code 'Hold Timer Expired'.
idChange **	*integer*	Number of notification messages with status code 'ID Change'. **Since: 4.21.3

`illegalCBit` **	*integer*	Number of notification messages with status code 'Illegal C Bit'. **Since: 4.21.3
`internalError`	*integer*	Number of notification messages with status code 'Internal Error'.
`ipChange` **	*integer*	Number of notification messages with status code 'IP Connectivity Change'. **Since: 4.21.3
`ipConnectivityChange` **	*integer*	Number of notification messages with status code 'IP Connectivity Change'. **Since: 4.23.0
`keepAliveExpired`	*integer*	Number of notification messages with status code 'KeepAlive Timer Expired'.
`labelRequestAborted`	*integer*	Number of notification messages with status code 'Label Request Aborted'.
`labelResourcesAvailable`	*integer*	Number of notification messages with status code 'Label Resources / Available'.
`linkAdjRemoved` **	*integer*	Number of notification messages with status code 'Link Adjacency Removed'. **Since: 4.26.2
`localTransportAddrChange` **	*integer*	Number of notification messages with status code 'Local Transport Address Change'. **Since: 4.26.2
`loopDetected`	*integer*	Number of notification messages with status code 'Loop Detected'.
`mLdpChange` **	*integer*	Number of notification messages with status code 'mLdp Change'. **Since: 4.22.0
`malformedTlv`	*integer*	Number of notification messages with status code 'Malformed TLV Value'.
`missingMessageParams`	*integer*	Number of notification messages with status code 'Missing Message Parameters'.
`neighborCleared` **	*integer*	Number of notification messages with status code 'Neighbor Cleared'. **Since: 4.26.2
`noAdjacency` **	*integer*	Number of notification messages with status code 'No Adjacency'. **Since: 4.26.2
`noLabelResources`	*integer*	Number of notification messages with status code 'No Label Resources'.
`noRoute`	*integer*	Number of notification messages with status code 'No Route'.
`peerAddrChange` **	*integer*	Number of notification messages with status code 'Peer Address Change'. **Since: 4.26.2
`pwStatus` **	*integer*	Number of notification messages with status code 'PW Status'. **Since: 4.21.3
`rejectMaxPduLength`	*integer*	Number of notification messages with status code 'Session Rejected / Parameters Max PDU Length'.

rejectNoHello	*integer*	Number of notification messages with status code 'Session Rejected / No Hello'.
rejectParamAdvMode	*integer*	Number of notification messages with status code 'Session Rejected / Parameters Advertisement Mode'.
rejectParamLabelRange	*integer*	Number of notification messages with status code 'Session Rejected / Parameters Label Range'.
sessionInitFailed **	*integer*	Number of notification messages with status code 'Session Init Failed'. **Since: 4.21.3
sessionRejectBadKeepAlive-Time	*integer*	Number of notification messages with status code 'Session Rejcted / Bad KeepAlive Time'.
sessionRoleChanged **	*integer*	Number of notification messages with status code 'Session Role Changed'. **Since: 4.21.3
shutdown	*integer*	Number of notification messages with status code 'Shutdown'.
success	*integer*	Number of notification messages with status code 'Success'.
targetAdjRemoved **	*integer*	Number of notification messages with status code 'Target Adjacency Removed'. **Since: 4.26.2
tcpConnectionClosed **	*integer*	Number of notification messages with status code 'Tcp Connection Closed'. **Since: 4.21.3
unknownFec	*integer*	Number of notification messages with status code 'Unknown FEC'.
unknownMessageType	*integer*	Number of notification messages with status code 'Unknown Message Type'.
unknownMessageTypeIgnore **	*integer*	Number of notification messages with status code 'Ignored Unknown Message Type'. **Since: 4.21.3
unknownTlv	*integer*	Number of notification messages with status code 'Unknown TLV'.
unsupportedAddressFamily	*integer*	Number of notification messages with status code 'Unsupported Address Family'.
wrongCBit **	*integer*	Number of notification messages with status code 'Wrong C Bit'. **Since: 4.21.3

2.1059.1.1.3 LdpSessionStateLogEntryModel

code	*enum: keepAliveExpired* | *badLdpId* | *unknown-MessageTypeIgnore* | *sessionRoleChanged* | *rejectParamLabelRange* | *noLabelResources* | *unknownMessageType* | *endOfLib* | *noAdjacency* | *authKeyChange* | *idChange* | *loopDetected* | *establishingTcpConnection* | *missingMessageParams* | *shutdown* | *tcpConnectionClosed* | *unknownFec* | *pwStatus* | *grConfigChange* | *badPduLength* | *badMessageLength* | *labelResourcesAvailable* | *targetAdjRemoved* | *unknownTlv* | *malformedTlv* | *sessionRejectBadKeep-AliveTime* | *rejectParamAdvMode* | *eolConfigChange* | *labelRequestAborted* | *neighborCleared* | *mLdpChange* | *holdTimerExpired* | *rejectMaxPduLength* | *illegalCBit* | *badTlvLength* | *wrongCBit* | *sessionInitFailed* | *badVersion* | *internalError* | *rejectNoHello* | *localTransportAddrChange* | *unsupportedAddressFamily* | *peerAddrChange* | *noRoute* | *linkAdjRemoved* | *success*	In states non-exist or expired, the code used for the last state transition.
reason	*string*	In states non-exist or expired, the reason describing the last state transition.
sessionState	*enum: stateInitialized* | *stateOpenRecv* | *stateOperational* | *stateExpired* | *stateUnknown* | *stateNonExist* | *stateOpenSent*	Ldp session state.
timeStamp	*floating point number*	Time of state change in seconds.

2.1060 show mpls ldp rcf [...]

see 2.194

2.1061 show mpls ldp tunnel [...]

revision 3

`entries`	*table(numeric string) = LdpTunnelTableEntry* see 2.1061.1	LDP tunnel table entries keyed by tunnel index.

2.1061.1 LdpTunnelTableEntry

`endpoint`	*IP address prefix*	Endpoint of the tunnel.
`vias`	*list of Via* see 2.151.1.1	List of nexthops.

2.1062 show mpls ldp tunnel mldp [...]

`entries`	*table(numeric string) = MldpTunnelTableEntry* see 2.1062.1	MLDP tunnel table entries keyed by tunnel index.

2.1062.1 MldpTunnelTableEntry

`mcastTreeId`	*string*	Multicast tree identifier.
`vias`	*list of Via* see 2.151.1.1	List of nexthops.

2.1063 show mpls lfib counters [...]

`counters`	*table(numeric string) = MplsLfibCounterLabelEntry* see 2.1063.1	A dictionary of ingress MPLS LFIB counters keyed by their top label.
`[multiLabelCounters]` **	*table(string) = MplsLfibCounterLabelEntry* see 2.1063.1	A dictionary of ingress multi-label MPLS LFIB counters keyed by their top labels (ordered top-most to bottom-most). **Since: 4.22.0

2.1063.1 MplsLfibCounterLabelEntry

[failedCounterAlloc] **	*boolean*	Counter resources for the label could not be allocated. **Since: 4.30.0
[source] **	*enum: bgpLu | isisSrV6Adj | staticMcast | srp2mp | rsvp | srtepolicy | isisSrToLdp | ldp | mldp | isisSrV4Prefix | bgpL3Vpn | bgpLuToIsisSr | ldpToIsisSr | debug | isisSrV6Prefix | isisSrToBgpLu | isisSrV6Area | pseudowire | gribi | isisSrV4Area | ldpToBgpLu | staticMpls | bgpL2Evpn | bgpLuToLdp | isisSrV4Adj*	LFIB source. **Since: 4.28.1
[sourceDetails] **	*string*	Details from the source for the label. **Since: 4.28.1
totalBytes	*integer*	Total number of received bytes on label.
totalPackets	*integer*	Total number of received packets on label.
unprogrammed	*boolean*	Route is unprogrammed in hardware.

2.1064 show mpls lfib counters no-resource

see 2.1063

2.1065 show mpls lfib counters summary

agreggateNoResourceBytes	*integer*	Total aggregated bytes for all MPLS LFIB label stacks which are unable to be individually counted due to lack of availble counter resources.
agreggateNoResourcePkts	*integer*	Total aggregated packets for all MPLS LFIB label stacks which are unable to be individually counted due to lack of availble counter resources.
[countableMplsLabels]	*integer*	Total unique MPLS label stacks being individually counted.
[noResourceMplsLabels]	*integer*	Total unique MPLS label stacks unable to be individuallycounted due to lack of availble counter resources.
totalMplsLabels	*integer*	Total unique MPLS label stacks in the LFIB.

2.1066 show mpls lfib input bgp labeled-unicast

revision 3

see 2.1049

2.1067 show mpls lfib route [...]

revision 3

see 2.1049

2.1068 show mpls parsing speculative

`entries`	*table(string) = boolean*	Status of MPLS speculative parsing, indexed by packet type.

2.1069 show mpls ping access-list [...]

see 2.53

2.1070 show mpls route [...]

revision 2

[allowDefaultRoute]	*boolean*	True if MPLS next-hop is allowed to resolve over default route.
mplsSupported	*boolean*	True if MPLS routing is supported.
[multiLabelRoutes] **	*table(string) = MplsHwRoute* see 2.1070.1	Dictionary of programmed multi-label routes, keyed by their incoming MPLS labels, comma-separated, ordered top-most to bottom-most. **Since: 4.22.0
routes	*table(numeric string) = MplsHwRoute* see 2.1070.1	Dictionary of programmed routes, keyed by their incoming MPLS label.

2.1070.1 MplsHwRoute

adjacencies	*table(numeric string) = MplsHwAdjacency* see 2.1070.1.1	Dictionary of adjacencies keyed by metric.
[adjacencySetUnprogrammed]	*boolean*	Adjacency set is programmed in hardware.
bestMetric	*integer*	Metric for the best available path.
label	*integer*	Incoming MPLS label, set to 1048576 (N/A) if labels attribute is present.
[labels] **	*string*	Incoming MPLS labels for multi-label routes, comma-separated, ordered top-most to bottom-most. **Since: 4.22.0
unprogrammed	*boolean*	Route is unprogrammed in hardware.
[viaSetType] **	*enum: unicast | multicast*	Via set type. **Since: 4.22.1

2.1070.1.1 MplsHwAdjacency

[backupIpLookupVias] **	*list of IpLookupVia* see 2.1049.1.1.5	List of backup IP lookup vias. **Since: 4.27.0
[backupMplsVias] **	*list of MplsHwVia* see 2.1070.1.1.1	List of backup MPLS vias. **Since: 4.22.1
[evpnEthernetSegmentFilterVias]	*list of EvpnEthernetSegmentFilterVia* see 2.1049.1.1.1	List of EVPN Ethernet Segment Filter vias.
[evpnVlanAwareFloodVias]	*list of EvpnVlanAwareFloodVia* see 2.1049.1.1.2	List of EVPN VLAN Aware Flood vias.
[evpnVlanAwareVias]	*list of EvpnVlanAwareVia* see 2.1049.1.1.3	List of EVPN VLAN Aware vias.
[evpnVlanFloodVias]	*list of EvpnVlanFloodVia* see 2.1049.1.1.4	List of EVPN VLAN Flood vias.
[ipLookupVias]	*list of IpLookupVia* see 2.1049.1.1.5	List of IP lookup vias.
[mplsVias]	*list of MplsHwVia* see 2.1070.1.1.1	List of MPLS vias.

[multicastIpLookupVias] **	*list of MulticastIpLookupVia* see 2.1049.1.1.7	List of Multicast IP lookup vias. **Since: 4.26.2
viaType	*enum: viaTypeMulticastIpLookup | viaTypeIpLookup | viaTypeMax | viaTypeIpLookupBackup | viaTypeEvpnVlanFlood | viaTypeEvpnEthernet-Segment | viaTypeVlan | viaTypeNexthopGroup | viaTypeEvpnVlanAware | viaTypeMplsIp | viaType-EvpnVlanAwareFlood | viaTypeExternalFec | viaTypeMplsUnknown | viaTypeMplsPseudowire*	Obsolete because an adjacency can contain mixed via types.
[vlanVias]	*list of VlanVia* see 2.1049.1.1.10	List of VLAN vias.

2.1070.1.1.1 MplsHwVia

[backupVias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of backup tunnel vias. **Since: 4.22.0
[dscpMode] **	*enum: pipe | platformDefault | undefinedTtlMode | uniform*	DSCP mode. **Since: 4.25.1
l2Intf	*interface name*	Egress L2 interface.
l3Intf	*interface name*	Egress L3 interface.
labelAction	*enum: unknown | swap | push | pop | forward*	Label action.
macAddr	*MAC address*	Destination MAC address.
[nexthopGroup] **	*string*	Nexthop Group for Via. **Since: 4.26.0
outLabel	*integer*	Out label for the packet.
payloadType	*enum: ipv4 | ipv6 | ethernet | mpls | undefinedPayload | autoDecide*	Type of payload being carried.
recursiveNextHop	*IP address*	Recursive nexthop IP address.
resolvedNextHop	*IP address*	Resolved nexthop IP address.
[skipEgressAcl]	*boolean*	True if egress ACL is bypassed for the inner IP packets after MPLS pop.
ttlMode	*enum: pipe | platformDefault | undefinedTtlMode | uniform*	TTL mode.
[tunnelDescriptor] **	*TunnelInfo* see 2.151.1.3.1	Tunnel information. **Since: 4.21.0
[vias] **	*list of TunnelViaInfo* see 2.151.1.3.1.2	List of nested tunnel vias. **Since: 4.21.0
vlanId	*integer*	VLAN ID.
[weight] **	*integer*	Weight of the Via. **Since: 4.26.0

2.1071 show mpls route summary

numBackupAdjs	*integer*	Number of backup adjacencies (via groups).
numHwAdjs	*integer*	Number of adjacencies (via groups) used in hardware.
numLabels	*integer*	Number of labels.
numUnprogrammedLabels	*integer*	Number of unprogrammed labels.

2.1072 show mpls rsvp

adminState	*enum: disabled	enabled*	Is RSVP enabled by config.
authentication	*enum: disabled	MD5*	Cryptographic authentication.
bypassTunnelCount	*integer*	Number of bypass tunnels.
egressCount	*integer*	Number of egress sessions.
fastReroute	*boolean*	Is fast re-route enabled.
fastRerouteMode	*enum: linkProtection	none	nodeProtection*	Fast re-route mode.
[grHelper] **	*GrHelper* see 2.1072.1	Graceful restart helper configuration. **Since: 4.29.0
[grSpeaker] **	*GrSpeaker* see 2.1072.2	Graceful restart speaker configuration. **Since: 4.29.0
helloEnabled	*boolean*	Are hello messages enabled.
[helloInterval]	*floating point number*	Time interval in seconds between RSVP hello messages. Set when helloEnabled is True.
[helloMultiplier]	*floating point number*	Number of missed hellos after which the neighbor is expired. Set when helloEnabled is True.
hierarchicalFecsEnabled	*boolean*	Are hierarchical FECs enabled.
[hitlessRestart] **	*HitlessRestart* see 2.1072.3	Hitless restart configuration. **Since: 4.30.1
ingressCount	*integer*	Number of ingress sessions.
interfaceCount	*integer*	Number of interfaces.
[labelLocalTerminationMode] **	*enum: implicitNull	explicitNull*	Label local termination mode. **Since: 4.24.1
lspCount	*LspCount* see 2.1072.4	LSP related counts.
mtuSignalingEnabled **	*boolean*	Is MTU signaling enabled. **Since: 4.23.2
neighborCount	*integer*	Number of neighbors.
operationalState	*enum: down	up*	Operational state of RSVP.
preemptionPeriod **	*floating point number*	Preemption timer value in seconds. **Since: 4.23.2
refreshInterval	*floating point number*	Time interval in seconds between RSVP neighbor state refresh messages.
refreshReduction	*boolean*	Is bundling of RSVP neighbor messages enabled.
[reversionMode]	*enum: global	local*	Operational FRR reversion mode.
[reversionModeConfig] **	*enum: global	local*	Configured FRR reversion mode. **Since: 4.22.1
sessionCount	*integer*	Number of sessions.
srlgMode **	*enum: srlgStrict	srlgLoose	srlgNone*	SRLG mode. **Since: 4.24.1
transitCount	*integer*	Number of transit sessions.

2.1072.1 GrHelper

[maximumAcceptedRecovery]	*floating point number*	Maximum accepted recovery value from neighbors in seconds.
[maximumAcceptedRestart]	*floating point number*	Maximum accepted restart value from neighbors in seconds.

2.1072.2 GrSpeaker

recovery	*floating point number*	Graceful Restart recovery value in seconds.
restart	*floating point number*	Graceful Restart restart value in seconds.

2.1072.3 HitlessRestart

recovery	*floating point number*	Hitless Restart recovery value in seconds.

2.1072.4 LspCount

egress	*integer*	Number of egress LSPs.
ingress	*integer*	Number of ingress LSPs.
operational	*integer*	Number of operational LSPs.
total	*integer*	Number of LSPs.
transit	*integer*	Number of transit LSPs.
usingBypassTunnels	*integer*	Number of LSP using bypass tunnels.

2.1073 show mpls rsvp ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.1074 show mpls rsvp bandwidth [...]

interfaces	*table(Interface name) = RsvpBandwidthIntfModel* see 2.1074.1	Interface's Bandwidth information.

2.1074.1 RsvpBandwidthIntfModel

reservedBandwidth	*table(numeric string) = floating point number*	A map of priority to bandwidth reserved in bits per second.
totalBandwidth	*floating point number*	Total bandwidth available to RSVP in bits per second.

2.1075 show mpls rsvp bandwidth summary [...]

interfaces	*table(Interface name) = RsvpBandwidthIntfSummaryModel* see 2.1075.1	A mapping of interface to bandwidth summary.

2.1075.1 RsvpBandwidthIntfSummaryModel

reservedBandwidth	*floating point number*	Bandwidth reserved in bits per second.
totalBandwidth	*floating point number*	Total bandwidth available to RSVP in bits per second.

2.1076 show mpls rsvp counters [...]

errorCounts **	*table(string) = RsvpErrorCounter* see 2.1076.1	Error counters. **Since: 4.27.2
interfaces	*table(Interface name) = RsvpInterfaceCounter* see 2.1076.2	Interface message counters.

2.1076.1 RsvpErrorCounter

rx	*integer*	Received error counts.
tx	*integer*	Sent error counts.

2.1076.2 RsvpInterfaceCounter

rx	*RsvpMessageCounter* see 2.1076.2.1	Received message counts.
tx	*RsvpMessageCounter* see 2.1076.2.1	Sent message counts.

2.1076.2.1 RsvpMessageCounter

counts	*table(string) = integer*	Message counts keyed by RsvpMessageType(Path, PathTear, PathErr, Resv, ResvTearSrefresh, Other, Errors).

2.1077 show mpls rsvp neighbor [...]

neighbors	*table(IPv4 or IPv6 address) = RsvpNeighbor* see 2.1077.1	Neighbor information.

2.1077.1 RsvpNeighbor

[authentication]	*enum: disabled | MD5*	Cryptographic authentication.
[bypassInfoNode] **	*table(IPv4 or IPv6 address) = NeighborBypassInfo* see 2.1077.1.1	A dictionary of bypass tunnel state for node protection keyed by tunnel destination IP address. **Since: 4.22.0
[bypassLabel]	*integer*	Bypass tunnel label.
[bypassNextHop]	*IP address*	Bypass tunnel next hop.
bypassTunnel	*enum: notAvailable | available | notRequested | inUse*	Bypass tunnel state.
[bypassTunnelName]	*string*	Bypass tunnel name.
downstreams	*list of NeighborLspInfo* see 2.1077.1.2	Downstream session LSP information.
[lastHelloReceived]	*floating point number*	UTC time when the last RSVP hello message was received.
[lastHelloSent]	*floating point number*	UTC time when the last RSVP hello message was sent.
[neighborCreationTime]	*floating point number*	UTC time when neighbor was created.
operGrState **	*enum: inactive | unknown | recovery | restart | unsupported*	Graceful restart state. **Since: 4.28.2
[recoveryTime] **	*floating point number*	Graceful restart recovery time advertised by the neighbor in seconds. **Since: 4.28.2
[restartTime] **	*floating point number*	Graceful restart restart time advertised by the neighbor in seconds. **Since: 4.28.2
upstreams	*list of NeighborLspInfo* see 2.1077.1.2	Upstream session LSP information.

2.1077.1.1 NeighborBypassInfo

[bypassLabel]	*integer*	Bypass tunnel label.
[bypassNextHop]	*IP address*	Bypass tunnel next hop.
bypassTunnel	*enum: notAvailable | available | notRequested | inUse*	Bypass tunnel state.
[bypassTunnelName]	*string*	Bypass tunnel name.

2.1077.1.2 NeighborLspInfo

lspId	*integer*	LSP ID.
sessionId	*integer*	Session ID.

`[sessionName]` **	*string*	Session name. **Since: 4.25.1

2.1078 show mpls rsvp neighbor summary

`neighborSummary`	*table(IPv4 or IPv6 address) = RsvpNeighborStatistics* see 2.1078.1	Neighbor summary.

2.1078.1 RsvpNeighborStatistics

`downstreamLspCount`	*integer*	Number of downstream LSPs.
`downstreamSessionCount`	*integer*	Number of downstream sessions.
`upstreamLspCount`	*integer*	Number of upstream LSPs.
`upstreamSessionCount`	*integer*	Number of upstream sessions.

2.1079 show mpls rsvp session [...]

revision 2

sessions	*table(numeric string) = RsvpSession* see 2.1079.1	Table of session information keyed by session number.

2.1079.1 RsvpSession

destination	*IP address*	Destination address.
[extendedTunnelId]	*IP address*	Extended tunnel ID.
lsps	*table(numeric string) = Lsp* see 2.1079.1.1	Table of LSP information keyed by LSP number.
[sessionCreationTime]	*floating point number*	UTC time when session was created.
[sessionHistory]	*RsvpHistory* see 2.1079.1.1.3	Session history events.
sessionRole	*enum: transit | egress | unknown | ingress*	Role for Session.
[state]	*enum: down | up*	Session operational state.
[tunnelId]	*integer*	Tunnel ID.

2.1079.1.1 Lsp

[backupDownstreamNeighborAddress]	*IP address*	Downstream backup neighbor address.
[backupUpstreamNeighborAddress]	*IP address*	Upstream backup neighbor address.
[bandwidth]	*floating point number*	Requested bandwidth in bps.
[bandwidthInterface] **	*interface name*	Bandwidth reservation interface. **Since: 4.29.1
[bandwidthState]	*enum: failedPreemption | failedAdmission | unknown | pending | preempted | reserved*	Current status of bandwidth reservation.
[bypassLabel]	*integer*	Bypass tunnel label.
[bypassLastHopLabel]	*integer*	Merge point label.
[bypassNextHop]	*IP address*	Bypass tunnel next hop.
bypassTunnel	*enum: notAvailable | egress | available | notRequested | inUse*	Bypass tunnel state.
[bypassTunnelName]	*string*	Bypass tunnel name.
[downstreamLabel]	*integer*	Downstream MPLS label.
[downstreamNeighbor]	*LspNeighborInfo* see 2.1079.1.1.1	Downstream neighbor information.
[dsRro] **	*list of RsvpRroHop* see 2.1079.1.1.4	Record route hops from downstream. **Since: 4.26.0
[ero] **	*list of RsvpEroHop* see 2.1079.1.1.2	Explicit route hops. **Since: 4.26.0
[frrOperationalMode]	*enum: linkProtection | none | nodeProtection*	Fast re-route mode operational.

[frrRequestedMode]	*enum: linkProtection | none | nodeProtection*	Fast re-route mode requested.
[holdPriority]	*integer*	Hold priority.
[inMtu]	*integer*	MTU received in the incoming path message.
[localLabel]	*integer*	Local MPLS label.
[lspCreationTime]	*floating point number*	UTC time when LSP was created.
[lspHistory]	*RsvpHistory* see 2.1079.1.1.3	SP history events.
[lspId]	*integer*	LSP ID.
[lspType]	*enum: bypass | primary*	LSP purpose.
[mtuSignalingEnabled]	*boolean*	MTU signaling is enabled for this LSP.
[outMtu]	*integer*	MTU received in the outgoing path message.
[path]	*list of string*	Explicit path.
pathState	*enum: sentOnly | receivedOnly | down | up*	Path state.
resvState	*enum: sentOnly | receivedOnly | down | up*	Reservation state.
[sessionName]	*string*	Head end session name.
[setupPriority]	*integer*	Setup priority.
[softPreemptionRequested]	*boolean*	Soft preemption has been requested.
[sourceAddress]	*IP address*	Source address.
state	*enum: down | up | pathOnly*	Operational state.
[unprogrammed] **	*boolean*	LSP is unprogrammed in hardware. **Since: 4.28.0
[upstreamNeighbor]	*LspNeighborInfo* see 2.1079.1.1.1	Upstream neighbor information.
[usRro] **	*list of RsvpRroHop* see 2.1079.1.1.4	Record route hops from upstream. **Since: 4.26.0

2.1079.1.1.1 LspNeighborInfo

address	*IP address*	Neighbor address.
[lastRefreshReceived]	*floating point number*	UTC time when the last neighbor refresh was received.
[lastRefreshSent]	*floating point number*	UTC time when the last neighbor refresh was sent.
[localInterface]	*interface name*	Local interface.

2.1079.1.1.2 RsvpEroHop

[asNumber]	*integer*	ERO hop AS number.
loose	*boolean*	ERO hop is a loose hop.
[prefix]	*IP address prefix*	ERO hop prefix.

2.1079.1.1.3 RsvpHistory

history	*list of RsvpHistoryEntry* see 2.1079.1.1.3.1	Recent history events.

2.1079.1.1.3.1 RsvpHistoryEntry

event	*string*	History event.
timestamp	*floating point number*	Timestamp of the event.

2.1079.1.1.4 RsvpRroHop

address	*IP address*	RRO hop address.
frrInUse	*boolean*	RRO hop actively uses fast re-route local protection.
frrMode	*enum: linkProtection | none | nodeProtection*	RRO hop's fast re-route mode.
[label]	*integer*	RRO hop label.
nodeId	*boolean*	RRO hop is a node ID.

2.1080 show mpls segment-routing bindings [...]

see 2.1048

2.1081 show mpls tunnel counters [...]

entries	*list of MplsTunnelCountersEntry* see 2.1081.1	MPLS tunnel hardware counters.

2.1081.1 MplsTunnelCountersEntry

[counterInUse] **	*boolean*	Counter is in use for the tunnel entry. **Since: 4.30.1
[endpoint]	*IP address prefix*	Endpoint of the tunnel.
tunnelIndex	*integer*	Index of the tunnel entry in the source tunnel table.
[tunnelName] **	*string*	Name for the tunnel entry. **Since: 4.26.2

tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.
[txBytes]	*integer*	Total number of transmitted bytes.
[txPackets]	*integer*	Total number of transmitted packets.
vias	*list of IpVia* see 2.151.1.2	List of nexthops.

2.1082 show mpls tunnel fib

revision 3

entries	*list of MplsTunnelFibEntry* see 2.1082.1	List of mpls tunnel FIB entries.

2.1082.1 MplsTunnelFibEntry

[endpoint]	*IP address prefix*	Endpoint of the tunnel.
tunnelIndex	*integer*	Index of the tunnel entry in the source tunnel table.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

[tunnelViaStatus] **	*enum: unresolved | usingBackupVias | unknown | usingPrimaryVias | notProgrammed*	Tunnel programming status. **Since: 4.21.3
vias	*list of MplsVia* see 2.151.1.3	List of nexthops.

2.1083 show mpls tunnel static [...]

revision 3

entries	*table(numeric string) = StaticTunnelTableEntry* see 2.1083.1	Static tunnel table entries keyed by tunnel index.

2.1083.1 StaticTunnelTableEntry

endpoint	*IP address prefix*	Endpoint of the tunnel.
name	*string*	Static tunnel name.
vias	*list of MplsVia* see 2.151.1.3	List of nexthops.

2.1084 show mpls tunnel termination qos maps [...]

vrfsToDscpToTcNamedMaps	*table(string) = string*	VRF to DSCP to TC named maps, indexed by VRF name.

2.1085 show msdp mesh-group [...]

see 2.606

2.1086 show msdp rpf-peer RP [...]

see 2.609

2.1087 show msrp [...]

interfaces	*table(Interface name) = Intf* see 2.1087.1	Msrp information for a port.
maxFanInPorts	*integer*	Maxiumum number of ingress ports supporting streaming down, beyond which no more Talker streams are allowed.
maxFrameSize	*integer*	Maximum allowed frame size.
msrpEnabled	*boolean*	Indicates whether Msrp is enabled.
srClasses	*table(string) = SrClassGlobal* see 2.1087.2	Globally configured parameters for SR Classes.

2.1087.1 Intf

operState	*enum: LinkDown | Disabled | Active | Blocked*	Operational Msrp state on this interface: Disabled – Msrp is disabled, LinkDown – Physical link is down, Blocked – STP port state is blocked, Active – Msrp is running successfully.
srClasses	*table(string) = SrClass* see 2.1087.1.1	Msrp information for a given SR Class.
srPvid	*integer*	Default VLAN ID for streams on this interface.

2.1087.1.1 SrClass

bandwidth	*integer*	Maximum bandwidth allowed for the SR Class.
bandwidthUnit	*enum: kbps | Mbps | bps | pps*	The unit of bandwidth for SR Classes.
listenerAskFailed	*integer*	Number of ListenerAskingFailed of a given SR Class received on the interface.
listenerReady	*integer*	Number of ListenerReady and ListenerReadyFailed of a given SR Class received on the interface.
state	*enum: WaitingForPeer | Boundary | Core | NotSupported*	The state of an SR Class.
talkerAdvertise	*integer*	Number of TalkerAdvertise of a given SR Class received on the interface.
talkerFailed	*integer*	Number of TalkerFailed of a given SR Class received on the interface.

2.1087.2 SrClassGlobal

bandwidth	*integer*	Maximum bandwidth allowed for SR Class.
priority	*integer*	Priority associated with the SR Class.
supported	*boolean*	Indicates whether SR Class is supported.

2.1088 show msrp counters [...]

interfaces	*table(Interface name) = Intf* see 2.1088.1	Msrp information for a port.

2.1088.1 Intf

operState	*enum: LinkDown | Disabled | Active | Blocked*	Operational Msrp state on this interface: Disabled – Msrp is disabled, LinkDown – Physical link is down, Blocked – STP port state is blocked, Active – Msrp is running successfully.
rxCounters	*table(string) = integer*	MSRP Rxcounters values for respective interfaces.
txCounters	*table(string) = integer*	MSRP TxCounters value for respective interfaces.

2.1089 show msrp interfaces (streams | listeners | talkers) [...]

interfaces	*table(Interface name) = Intf* see 2.1089.1	Msrp streams information for a given interface.

2.1089.1 Intf

streamListenerData	*table(string) = Listener* see 2.1089.1.1	Msrp listener information for a given stream.
streamTalkerData	*table(string) = Talker* see 2.1089.1.2	Msrp talker information for a given stream.

2.1089.1.1 Listener

listenerAge	*string*	Age since the last state of Listener changed.
listenerDirection	*enum: Tx | Rx | None*	Rx – Listener message was received, Tx – Listener message was transmitted, None – No Listener present.
listenerState	*enum: NoListener | ListenerAskingFailed | ListenerReady | ListenerReadyFailed*	Listener state on the given stream.

2.1089.1.2 Talker

talkerAge	*string*	Age since the last state of Talker changed.
talkerDirection	*enum: Tx | Rx | None*	Rx – Talker message was received, Tx – Talker message was transmitted, None – No Talker present.

talkerFailureCode	*enum: insufficientTrafficClassBandwidth* | *firstValueChangedForStreamId* | *egressPortNotAvbCapable* | *latencyHasChanged* | *srClassPriorityMismatch* | *priorityIsNotAnSrClass* | *noFailure* | *outOfMsrpResources* | *maxFanInPortsLimitReached* | *maxFrameSizeTooLarge* | *streamPreemptedByHigherRank* | *insufficientResources* | *insufficientBandwidth* | *outOfMmrpResources* | *useDifferentDestinationAddress* | *streamDestinationAddressInUse* | *streamIdInUse* | *vlanTaggingDisabledOnEgress* | *cannotStoreDestinationAddress* | *vlanBlockedOnEgress*	Error code behind the Talker Failed message.
talkerState	*enum: TalkerFailed* | *TalkerAdvertise* | *NoTalker*	Talker state on the given stream.

2.1090 show msrp mrp [...]

interfaces	*table(Interface name) = IntfDatabase* see 2.1090.1	MRP attribute information indexed by interface.

2.1090.1 IntfDatabase

attributeTypes	*table(numeric string) = AttrTypeDatabase* see 2.1090.1.1	MRP type attribute information.

2.1090.1.1 AttrTypeDatabase

attributes	*table(string) = Attribute* see 2.1090.1.1.1	MRP attribute information indexed by keys.

2.1090.1.1.1 Attribute

applicantState	*enum: mrpAnxiousNew* | *mrpVeryAnxiousPassive* | *mrpAnxiousObserver* | *mrpVeryAnxiousNew* | *mrpAnxiousActive* | *mrpQuietActive* | *mrpAnxiousPassive* | *mrpQuietObserver* | *mrpQuietPassive* | *mrpLeavingActive* | *mrpVeryAnxiousObserver* | *mrpLeavingObserver*	MRP applicant state.
registrarState	*enum: mrpLv* | *mrpMt* | *mrpIn*	MRP registrar state.
[value]	*string*	String representation of the MRP attribute value.

2.1091 show msrp streams [...]

streams	*table(string) = Stream* see 2.1091.1	Msrp stream information collected from the Talker attributes.

2.1091.1 Stream

bandwidth	*integer*	Bandwidth allocated to the stream.
bandwidthUnit	*enum: kbps* | *Mbps* | *bps* | *pps*	The unit of bandwidth for SR Classes.
[bridgeFailureCode]	*enum: insufficientTrafficClassBandwidth* | *firstValueChangedForStreamId* | *egressPortNotAvbCapable* | *latencyHasChanged* | *srClassPriorityMismatch* | *priorityIsNotAnSrClass* | *noFailure* | *outOfMsrpResources* | *maxFanInPortsLimitReached* | *maxFrameSizeTooLarge* | *streamPreemptedByHigherRank* | *insufficientResources* | *insufficientBandwidth* | *outOfMmrpResources* | *useDifferentDestinationAddress* | *streamDestinationAddressInUse* | *streamIdInUse* | *vlanTaggingDisabledOnEgress* | *cannotStoreDestinationAddress* | *vlanBlockedOnEgress*	Failure code associated with bridge that changed Talker Advertise to Talker Failed.
destinationMacAddr	*MAC address*	Stream destination MAC address.

[failedBridgeId]	*integer*	Bridge ID of Bridge that changed Talker Advertise to Talker Failed.
[ingressIntf]	*interface name*	Interface on which talker has registered the stream.
[latency]	*integer*	Accumulated latency in nanoseconds for the stream.
[maxFrameSize]	*integer*	Maximum frame size sent by talker.
[maxIntervalFrames]	*integer*	Maximum number of frames sent per class measurement.
[srClass]	*enum: SrClassB* | *SrClassA*	SR Class of the stream.
[talkerState]	*enum: TalkerFailed* | *TalkerAdvertise* | *NoTalker*	State of the talker on the ingress interface.
[vlanId]	*integer*	Vlan Id for the stream.

2.1092 show msrp streams count

activeAvbStreams	*integer*	Number of established AVB Streams. An established AVB stream is one which has a registered talker and has bandwidth allocated to it.

2.1093 show msrp streams propagation [...]

streams	*table(string) = Stream* see 2.1093.1	Shows the propagation of talker and listener attributes for a given stream through the switch.

2.1093.1 Stream

bandwidth	*integer*	Bandwidth allocated to the stream.
bandwidthUnit	*enum: kbps | Mbps | bps | pps*	The unit of bandwidth for SR Classes.
destinationMacAddr	*MAC address*	Stream destination MAC address.
[listenerEgressIntf]	*interface name*	Egress interface on which the listener message was transmitted.
listenerIngressInterfaces	*table(Interface name) = ListenerIngressPort* see 2.1093.1.1	Ingress interfaces on which the listener attributes for a given stream are received.
listenerStateEgressIntf	*enum: NoListener | ListenerAskingFailed | ListenerReady | ListenerReadyFailed*	State of the listener in the message transmitted on the egress interface.
[srClass]	*enum: SrClassB | SrClassA*	SR Class of the stream.
talkerEgressInterfaces	*table(Interface name) = TalkerEgressPort* see 2.1093.1.2	Egress interfaces on which the talker state of a given stream is propagated.
[talkerIngressIntf]	*interface name*	Interface on which the talker message was received.
talkerStateIngressIntf	*enum: TalkerFailed | TalkerAdvertise | NoTalker*	State of the talker in the message received on the ingress interface.
talkerStatePropagated	*enum: TalkerFailed | TalkerAdvertise | NoTalker*	State of the talker propagated on the given egress interface.
[vlanId]	*integer*	Vlan Id for the stream.

2.1093.1.1 ListenerIngressPort

listenerStateIngress	*enum: NoListener | ListenerAskingFailed | ListenerReady | ListenerReadyFailed*	State of the listener in the message received on the given ingress interface.
[listenerStatePropagated]	*enum: NoListener | ListenerAskingFailed | ListenerReady | ListenerReadyFailed*	State of the listener propagated to the egress interface.

2.1093.1.2 TalkerEgressPort

talkerStateEgress	*enum: TalkerFailed | TalkerAdvertise | NoTalker*	State of the talker propagated on the given egress interface.

2.1094 show multicast (ipv4 | ipv6) evpn decap joined [...]

interfaces	*table(string) = MrouteVxlanDecapInterface* see 2.1094.1	A mapping of interface to joined groups.

2.1094.1 MrouteVxlanDecapInterface

groups	*table(IPv4 or IPv6 address) = MrouteVxlanDecapGroup* see 2.1094.1.1	A mapping of group to it's sources.

2.1094.1.1 MrouteVxlanDecapGroup

sources	*list of IP address*	List of sources for this group.

2.1095 show multicast (ipv4 | ipv6) source route export [...]

revision 2

see 2.45

2.1096 show multicast AF evpn decap received

[underlayReplication] **	*UnderlayReplicationModel* see 2.1096.2	Underlay replication. **Since: 4.28.1
vlans	*table(numeric string) = MulticastTunnelVlan* see 2.1096.1	A mapping of VLAN ID to multicast tunnel encapsuslation information.

2.1096.1 MulticastTunnelVlan

groups	*table(IPv4 or IPv6 address) = MulticastTunnelGroup* see 2.1096.1.1	A mapping of a multicast group in a VLAN to traffic sources.

2.1096.1.1 MulticastTunnelGroup

sources	*table(IPv4 or IPv6 address) = MulticastTunnelSource* see 2.1096.1.1.1	A mapping of source to encapsulating multicast sources and groups.

2.1096.1.1.1 MulticastTunnelSource

groups	*list of MulticastEncapSourceGroup* see 2.1096.1.1.1.1	A list of encapsulating groups for this (S,G).

2.1096.1.1.1.1 MulticastEncapSourceGroup

`group`	*IP address*	Destination group address used in tunnel encapsulation.
`source`	*IP address*	Source address used in tunnel encapsulation.
`[tunnelType]`	*enum: pimSsm | pimBidir | pimSm*	Tunnel type.

2.1096.2 UnderlayReplicationModel

`headendReplication`	*boolean*	Underlay HER enabled.
`multicast`	*boolean*	Underlay multicast enabled.

2.1097 show multicast AF evpn decap received flood

`vlans`	*table(numeric string) = MulticastFloodVlan* see 2.1097.1	A mapping of VLAN ID to multicast tunnel encapsuslation information for broadcast, unknown unicast and link-local multicast traffic.

2.1097.1 MulticastFloodVlan

`floodGroups`	*list of MulticastEncapSourceGroup* see 2.1096.1.1.1.1	A list of encapsulating (S,G) for a VLAN.

2.1098 show multicast AF evpn encap

see 2.1096

2.1099 show multicast AF evpn encap flood

see 2.1097

2.1100 show multicast fib ipv4 PIM_MODE [...]

see 2.594

2.1101 show multicast fib ipv4 [...]

see 2.594

2.1102 show multicast fib ipv4 df [...]

see 2.596

2.1103 show multicast fib ipv4 rpa [...]

see 2.597

2.1104 show multicast fib ipv6 PIM_MODE [...]

see 2.594

2.1105 show multicast fib ipv6 [...]

see 2.594

2.1106 show multicast ipv4 boundary [...]

see 2.612

2.1107 show multicast ipv4 evpn encap mapping (vrf VRF) OVERLAY_GRP

`vlans`	*table(numeric string) = IP address*	A mapping of VLAN ID to multicast tunnel encapsuslation group.

2.1108 show multicast ipv6 boundary [...]

see 2.612

2.1109 show mvpn ipv4 vrf

`vrfs`	*table(string) = MvpnVrfPmsis* see 2.1109.1	Mapping of VRF to PMSIs.

2.1109.1 MvpnVrfPmsis

`pmsis`	*list of string*	List of PMSIs for MVPN VRF.

2.1110 show mvrp [...]

interfaces	*table(Interface name) = Intf* see 2.1110.1	Mvrp information for a port.
mvrpEnabled	*boolean*	Indicates whether Mvrp is enabled.

2.1110.1 Intf

declaredVlans	*list of integer*	Mvrp vlans declared on a given port.
operState	*enum: LinkDown | Disabled | Active | Blocked*	Operational Mvrp state on this interface: Disabled – Mvrp is disabled, LinkDown – Physical link is down, Blocked – STP port state is blocked, Active – Mvrp is running successfully.
registeredVlans	*list of integer*	Mvrp vlans registered on a given port.

2.1111 show mvrp mrp [...]

interfaces	*table(Interface name) = IntfDatabase* see 2.1111.1	MRP attribute information indexed by interface.

2.1111.1 IntfDatabase

attributeTypes	*table(numeric string) = AttrTypeDatabase* see 2.1111.1.1	MRP type attribute information.

2.1111.1.1 AttrTypeDatabase

attributes	*table(numeric string) = Attribute* see 2.1111.1.1.1	MRP attribute information indexed by keys.

2.1111.1.1.1 Attribute

applicantState	*enum: mrpAnxiousNew | mrpVeryAnxiousPassive | mrpAnxiousObserver | mrpVeryAnxiousNew | mrpAnxiousActive | mrpQuietActive | mrpAnxiousPassive | mrpQuietObserver | mrpQuietPassive | mrpLeavingActive | mrpVeryAnxiousObserver | mrpLeavingObserver*	MRP applicant state.
registrarState	*enum: mrpLv | mrpMt | mrpIn*	MRP registrar state.

[value]	*string*	String representation of the MRP attribute value.

2.1112 show mvrp vlans [...]

vlans	*table(numeric string) = Vlan* see 2.1112.1	Mvrp information for a vlan.

2.1112.1 Vlan

declaredIntfs	*list of string*	Ports on which a given Mvrp Vlan is declared.
registeredIntfs	*list of string*	Ports on which a given Mvrp Vlan is registered.

2.1113 show nat map-t domain [...]

domains	*table(string) = Domain* see 2.1113.1	MAP domains, keyed by name.

2.1113.1 Domain

brPrefix	*IP address prefix*	Border Relay IPv6 prefix.
eaLength	*integer*	Length of the Embedded Address bits.
exceptionOffloadIp	*IPv4 address*	Exception offload nexthop IP address.
exceptionOffloadState	*enum: unresolvedDrop | resolved | unconfiguredDrop*	State of the exception offload next hop.
[exceptionOffloadToCpu] **	*boolean*	Exception offload using local CPU. **Since: 4.30.0
mapId	*integer*	MAP domain ID.
mtu	*integer*	MTU for traffic headed to the MAP domain.
psidOffset	*integer*	Port Set IDentifier offset within the L4 port.
rules	*table(IPv4 or IPv6 address prefix) = Rule* see 2.1113.1.1	A mapping of IPv6 prefix to MAP rule.
state	*enum: hwProgrammingFailed | unknown | complete | programmedInHw | incomplete*	State of the domain.
vrf	*string*	VRF name.

2.1113.1.1 Rule

description	*string*	Description of the MAP rule.
ipv4Prefix	*IP address prefix*	IPv4 rule prefix.

ipv6Prefix	*IP address prefix*	IPv6 rule prefix.

2.1114 show nat map-t domain counters [...]

domains	*table(string) = DomainCountersSummary* see 2.1114.1	A mapping of MAP domain name to MAP domain counters.

2.1114.1 DomainCountersSummary

drop	*PktsAndBytes* see 2.1114.1.1	A cumulative count of dropped traffic in a MAP domain.
fourToSix	*PktsAndBytes* see 2.1114.1.1	A cumulative count of all 4To6 translated traffic in a MAP domain.
offload	*PktsAndBytes* see 2.1114.1.1	A cumulative count of all offloaded traffic in a MAP domain.
sixToFour	*PktsAndBytes* see 2.1114.1.1	A cumulative count of all 6To4 translated traffic in a MAP domain.

2.1114.1.1 PktsAndBytes

byteCount	*integer*	Byte count.
packetCount	*integer*	Packet count.

2.1115 show nat map-t domain counters drop [...]

domains	*table(string) = DomainDropCounters* see 2.1115.1	A mapping of MAP domain name to MAP domain drop counters.

2.1115.1 DomainDropCounters

spoofCheckFailed	*PktsAndBytes* see 2.1114.1.1	A packet/bytes count of all traffic that failed the spoof checks.

2.1116 show nat map-t domain counters exception [...]

domains	*table(string) = DomainExceptionCounters* see 2.1116.1	A mapping of MAP domain name to MAP domain exception counters.

2.1116.1 DomainExceptionCounters

fourToSixFragments-Reassembled	*PktsAndBytes* see 2.1114.1.1	A count of 4To6 reassembled fragments.
fourToSixFragmentsReceived	*PktsAndBytes* see 2.1114.1.1	A count of 4To6 received fragments.
fourToSixFragments-Transmitted	*PktsAndBytes* see 2.1114.1.1	A count of 4To6 transmitted fragments.

fourToSixPacketsFragmented	*PktsAndBytes* see 2.1114.1.1	A count of 4To6 fragmented packets.
sixToFourFragments-Reassembled	*PktsAndBytes* see 2.1114.1.1	A count of 6To4 reassembled fragments.
sixToFourFragmentsReceived	*PktsAndBytes* see 2.1114.1.1	A count of 6To4 received fragments.

2.1117 show nat map-t domain counters offload [...]

domains	*table(string) = DomainOffloadCounters* see 2.1117.1	A mapping of MAP domain name to MAP domain offload counters.

2.1117.1 DomainOffloadCounters

ipv4Fragment	*PktsAndBytes* see 2.1114.1.1	A count of all IPv4 traffic that was offloaded for reassembly.
ipv4MtuExceeded	*PktsAndBytes* see 2.1114.1.1	A count of all traffic that was offloaded for fragmentation.
ipv4NonTcpUdp	*PktsAndBytes* see 2.1114.1.1	A count of all non-TCP/UDP IPv4 traffic that was offloaded.
ipv4Options	*PktsAndBytes* see 2.1114.1.1	A count of all IPv4 options traffic that was offloaded.
ipv4UdpChecksum0	*PktsAndBytes* see 2.1114.1.1	A count of all traffic that was offloaded for checksum recalc.
ipv6Fragment	*PktsAndBytes* see 2.1114.1.1	A count of all IPv6 traffic that was offloaded for reassembly.
ipv6NonTcpUdp	*PktsAndBytes* see 2.1114.1.1	A count of all non-TCP/UDP IPv6 traffic that was offloaded.

2.1118 show nat map-t domain counters rule [...]

domains	*table(string) = DomainRuleCounters* see 2.1118.1	A mapping of MAP domain name to rule counters for that domain.

2.1118.1 DomainRuleCounters

rules	*table(IPv4 or IPv6 address prefix) = RuleCounters* see 2.1118.1.1	Mapping of IPv6 prefix to MAP rule.

2.1118.1.1 RuleCounters

fourToSix	*PktsAndBytes* see 2.1114.1.1	A packet/byte count of all 4To6 traffic for this rule.
ipv4Prefix	*IP address prefix*	IPv4 rule prefix.
ipv6Prefix	*IP address prefix*	IPv6 rule prefix.
sixToFour	*PktsAndBytes* see 2.1114.1.1	A packet/byte count of all 6To4 traffic for this rule.

2.1119 show nat map-t summary

domains	*table(string) = DomainSummary* see 2.1119.1	A mapping of domain name to its summary.

2.1119.1 DomainSummary

mapId	*integer*	MAP domain ID.
ruleCount	*integer*	Count of rules in this domain.
state	*enum: hwProgrammingFailed | unknown | complete | programmedInHw | incomplete*	State of the domain.

2.1120 show neighbor classification [...]

neighbors	*table(MAC address) = NbrClassificationModel* see 2.1120.1	A dictionary of neighbors classifications keyed by ethernet address.

2.1120.1 NbrClassificationModel

classification	*list of string*	Classifications.
description	*string*	Description that helps identify neighbor.
interface	*interface name*	Interface that neighbor is connected to.
source **	*string*	Source of phone neighbor. **Since: 4.29.2

2.1121 show neighbor classification summary [...]

none	*NbrClassificationSummaryModel* see 2.1121.1	Summary for devices classified as none.
phone	*NbrClassificationSummaryModel* see 2.1121.1	Summary for devices classified as phone.

2.1121.1 NbrClassificationSummaryModel

aaa	*integer*	Deprecated - Number of AAA neighbors.
cli	*integer*	Number of CLI (manual) neighbors.
lldp	*integer*	Number of LLDP neighbors.

2.1122 show network physical-topology hosts [...]

hosts	*table(string) = Host* see 2.1122.1	Set of hosts in topology.

2.1122.1 Host

[details]	*boolean*	Details including logical ports and port channels.
hostname	*string*	Hostname of the device. This is typically configured.
logicalPorts	*table(string) = LogicalPort* see 2.1122.1.1	Logical ports on this host.
name	*string*	Unique name/id which identifies this device.
port	*table(string) = Port* see 2.1122.1.2	Ports on this host.
portChannels	*table(string) = PortGroup* see 2.1122.1.3	Port channels on this host.

2.1122.1.1 LogicalPort

[memberPorts]	*list of string*	Names of the member ports in this port channel.
name	*string*	Name of the logical port, port channel or physical port.

2.1122.1.2 Port

hostid **	*string*	Unique name/id which identifies host owning this port. **Since: 4.20.0
hostname	*string*	Host owning this port.
name	*string*	Name of the port, for e.g., Ethernet1, Management1..
[portChannel]	*string*	Port channel which this port is a member of.

2.1122.1.3 PortGroup

memberPorts	*list of string*	Names of the member ports in this port channel.
name	*string*	Name of the portGroup, for e.g., portChannel1.

2.1123 show network physical-topology neighbors [...]

neighbors	*table(string) = DirectedEdge* see 2.1123.1	Collection of neighbors in topology.

2.1123.1 DirectedEdge

fromPort	*Port* see 2.1122.1.2	Starting port of the directed edge.
toPort	*list of Port* see 2.1122.1.2	Ending port of the directed edge.

2.1124 show nexthop-group [...]

nexthopGroups	*table(string) = NexthopGroup* see 2.1124.1	Mapping between nexthop group name and status.

2.1124.1 NexthopGroup

[autoSize] **	*boolean*	Nexthop group auto resizing. **Since: 4.15.3
destinationIps	*table(numeric string) = string*	Destination Ip address list.
[entryCounterType] **	*enum: shared | unshared*	Entry counter type. **Since: 4.20.5
[greKeyType]	*enum: noGreKey | customKey | ingressIntf*	GRE key type.
[hierarchicalFecsEnabled] **	*boolean*	True if hierarchical fecs are enabled for this nexthop group. **Since: 4.21.0
nexthopGroupId	*integer*	Nexthop group Id.
nexthopGroupName	*string*	Nexthop group name.
nexthopGroupType	*enum: vxlan | dzgre | invalid | ipv4OverUdp | ipv6OverUdp | ipInIp | mpls | ip | gre | mplsOverGre | mplsOverUdp*	Nexthop group type.
[numProgrammedTunnels] **	*integer*	Number of tunnels programmed in hardware. **Since: 4.15.3
[persistentCounters] **	*boolean*	Entry counters persist through configuration updates. **Since: 4.29.1
[programmed]	*boolean*	Whether the group has been programmed in hardware.
size	*integer*	Nexthop group entry size.
[sourceIntf] **	*interface name*	Source interface. **Since: 4.17.0
sourceIp	*IP address*	Source Ip address.
[tos] **	*integer*	Type of service value. **Since: 4.25.2
ttl	*integer*	Time to live value.
via	*table(numeric string) = Via* see 2.1124.1.1	Mapping between entry index, and via.

2.1124.1.1 Via

[bytes]	*integer*	Number of bytes matched.
[counterInUse] **	*boolean*	True if counter is in use. **Since: 4.30.1
destinationAddr	*IP address*	Destination Ip address.
entryIndex	*integer*	Nexthop group entry index.
isTunnelVia	*boolean*	True if destination is a tunnel.

l2Via	*L2Via* see 2.1124.1.1.1	L2 via.
l3Intf	*interface name*	Egress L3 interface.
[mplsLabelOperation] **	*enum: swap | push | pop | unknown*	Label Stack Operation. **Since: 4.22.0
[mplsLabelStack]	*list of integer*	MPLS label stack to push on to the packet. Left most label is the top of the stack.
[nextLevelFecIndex] **	*integer*	Next level FIB FEC if the via is hierarchical. **Since: 4.21.0
[nextLevelNexthopGroupName]	*string*	Next level nexthop group name.
nexthopAddr	*IP address*	Nexthop address.
[nexthopGroupType]	*enum: vxlan | dzgre | invalid | ipv4OverUdp | ipv6OverUdp | ipInIp | mpls | ip | gre | mplsOverGre | mplsOverUdp*	Nexthop group type.
[packets]	*integer*	Number of packets matched.
[platformNextLevelFecIndex] **	*integer*	Platform-specific version of the next level FIB FEC. **Since: 4.21.0
route	*IP address prefix*	Route used to resolve destination.
[sharedTunnel]	*boolean*	True for shared tunnel.
[sourceIp] **	*IP address*	Tunnel Source IP. **Since: 4.29.0
[unresolvedNexthopAction] **	*enum: drop | notProgrammed*	Nexthop programming action if not resolved. **Since: 4.15.3
[unresolvedNexthopReason]	*string*	A description of why this next hop is unresolved.

2.1124.1.1.1 L2Via

l2Intf	*interface name*	Egress L2 interface.
macAddr	*MAC address*	Destination MAC address.
vlanId	*integer*	Vlan Id.

2.1125 show nexthop-group id [...]

nexthopGroups	*table(numeric string) = NexthopGroup* see 2.1124.1	Mapping between nexthop group ID and status.

2.1126 show nexthop-group id mapping

nexthopGroups	*table(numeric string) = string*	Mapping between nexthop group ID and name.

2.1127 show nexthop-group summary

numNexthopGroups	*integer*	Number of nexthop groups configured.
numNexthopGroupsPerSize	*table(string) = integer*	Number of nexthop groups configured indexed by number of entries.
numNexthopGroupsPerType	*table(string) = integer*	Number of nexthop groups configured indexed by nexthop group type.
numUnprogrammedNexthop-Groups	*integer*	Number of unprogrammed nexthop groups.

2.1128 show nexthop-group tunnel [...]

indexes	*table(numeric string) = NexthopGroupTunnelTable-Entry* see 2.1128.1	Nexthop Group tunnel entries keyed by tunnel index.

2.1128.1 NexthopGroupTunnelTableEntry

endpoint	*IP address prefix*	Endpoint of the tunnel.
igpMetric	*integer*	IGP Metric for tunnel entry.
igpPref	*integer*	IGP Preference for tunnel entry.
nexthopGroupName	*string*	Nexthop group name.

2.1129 show ntp ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.1130 show ntp associations

peers	*table(string) = NtpPeer* see 2.1130.1	A mapping between peer host name and peer host informaion.

2.1130.1 NtpPeer

condition	*enum: backup | falsetick | reject | candidate | excess | outlier | pps.peer | sys.peer*	Select field of the peer status word.
delay	*floating point number*	Roundtrip delay (milliseconds).
jitter	*floating point number*	jitter (milliseconds).
lastReceived	*floating point number*	Timestamp of last received packet.
offset	*floating point number*	offset of server relative to this host (milliseconds).
peerIpAddr	*IP address*	IP of the peer.
peerType	*enum: unicast | broadcast | local*	Type of peer.
pollInterval	*integer*	Poll interval (seconds).
reachabilityHistory	*list of boolean*	A list of poll history success.

refid	*string*	Association ID or KISS code.
stratumLevel	*integer*	Accuracy of the clock source (1 is best).

2.1131 show ntp status

[maxEstimatedError]	*integer*	Maximum estimated error of the clock (in milliseconds).
[pollingInterval]	*integer*	Polling interval of the peer (in seconds).
[server]	*IP address*	Time source (NTP server or reference clock) to which the system clock is currently synchronised.
status	*enum: starting* | *disabled* | *unsynchronised* | *synchronised* | *non-existing*	Status of NTP synchronisation service.
[stratum]	*integer*	Indicates the level of the server in the NTP hierarchy - see RFC 5905, 7.3.

2.1132 show openflow

bindMode	*BindModeEntries* see 2.1132.1	Bind mode.
connectSuccessCount	*integer*	Connect success count.
[connectedCtrl]	*list of ConnectedController* see 2.1132.2	List of connected controllers.
controllers	*list of IpAddrAndPort* see 2.616.1.1.1.1	List of openflow controllers.
[controllersInfo]	*list of ControllerConfig* see 2.1132.3	List of openflow controllers.
datapathID	*integer*	Datapath ID.
description	*string*	Datapath description.
enabled	*boolean*	OpenFlow enabled.
[expiredTtlOutput-Controller]	*boolean*	Send expired ttl packets to controller.
flowTableProfile	*enum: vxlan-match* | *full-match* | *auto* | *l2-match*	Table profile.
flowTableState	*boolean*	Flow table enabled or not.
forwardingPipeline	*enum: flow-and-normal* | *flow*	Forwarding pipeline.
ipRoutingState	*boolean*	Enabled if rvlans.
keepalivePeriod	*integer*	Keepalive in seconds.
[minimumVersion]	*1.3*	Minimum OpenFlow protocol version required.
[passiveControllerConfig]	*list of ControllerConfig* see 2.1132.3	List of configured passive controllers.
[passiveControllerStatus]	*list of ConnectedController* see 2.1132.2	List of listening passive controllers.
[rvlans]	*RvlanEntries* see 2.1132.4	rvlan entries.
shellCommandAllowed	*boolean*	Shell commands allowed.
totalPacketCount	*integer*	Matched packets.

2.1132.1 BindModeEntries

[bindInterfaces]	*list of Interface name*	Bind interfaces.
bindMode	*enum: bindModeInterface* | *bindModeMonitor* | *bindModeVlan* | *None*	Bind mode.
[bindVlan]	*string*	Bind vlans.
[nativeVlan]	*string*	Native vlan.

2.1132.2 ConnectedController

auxConnected	*boolean*	Auxiliary channel connected.
controllerAddr	*IpAddrAndPort* see 2.616.1.1.1.1	Connected controller.
[controllerRole]	*enum: master* | *equal* | *slave*	Controller role.
[negotiatedVersion]	*string*	Version negotiated with the connected controller.

2.1132.3 ControllerConfig

auxEnabled	*boolean*	Auxiliary channel connected.
connectionMode	*enum: passive | active*	Switch's network connection mode with controller.
controllerAddr	*IpAddrAndPort* see 2.616.1.1.1.1	Connected controller.

2.1132.4 RvlanEntries

rIntf	*interface name*	Recirculation interface.
rvlanEntry	*list of Entries* see 2.1132.4.1	rvlan entries.

2.1132.4.1 Entries

rvlan	*integer*	rvlan entry.
vlan	*integer*	Vlan entry.

2.1133 show openflow flows [...]

flows	*list of Flow* see 2.296.1.1	Flows.

2.1134 show openflow groups [...]

groups	*list of Group* see 2.1134.1	Groups.

2.1134.1 Group

[buckets]	*list of Bucket* see 2.1134.1.1	List of buckets in the group.
flows	*list of string*	Flows referring to the group.
groupId	*integer*	Group Identifier.
status	*string*	Status of the group.

2.1134.1.1 Bucket

bucketId	*integer*	Bucket Id.
[macDst]	*MacAddr* see 2.296.1.1.1.3	Set destination Mac address.
[macSrc]	*MacAddr* see 2.296.1.1.1.3	Set source Mac address.
[outInterfaces]	*list of Interface name*	Output interfaces.
[ttlDec]	*boolean*	Decrement TTL.

[vlan]	*integer*	Set Vlan ID.

2.1135 show openflow ports

portInterface	*list of Entry* see 2.1135.1	List of openflow Ports to switch interface.

2.1135.1 Entry

intf	*interface name*	Switch interface.
mirror	*boolean*	mirror destination.
port	*integer*	OpenFlow port.

2.1136 show openflow profiles

tables	*list of OpenFlowTableProfile* see 2.1136.1	Table profiles.

2.1136.1 OpenFlowTableProfile

action	*OpenFlowTableProfile-Action* see 2.1136.1.1	Actions.
match	*OpenFlowTableProfile-Match* see 2.1136.1.2	Match conditions.
numFlowEntries	*integer*	Table size entries.
profile	*string*	Table profile.
wildcardMatch	*OpenFlowTableProfile-Match* see 2.1136.1.2	Wildcard match conditions.

2.1136.1.1 OpenFlowTableProfileAction

egrMirrorInterfaces	*boolean*	Copy egress to mirror destination interfaces.
icmpCode **	*boolean*	Set ICMP code. **Since: 4.23.1
icmpType **	*boolean*	Set ICMP type. **Since: 4.23.1
ingrMirrorInterfaces	*boolean*	Copy ingress to mirror destination interfaces.
ipDst	*boolean*	Set destination IP address.
ipPortDst	*boolean*	Set TCP/UDP destination port.
ipPortSrc	*boolean*	Set TCP/UDP source port.
ipSrc	*boolean*	Set source IP address.
ipTos	*boolean*	Set Ipv4 Tos.
loopback	*boolean*	Output to ingress interface.
macDst	*boolean*	Set destination Mac address.
macSrc	*boolean*	Set source Mac address.
outInterfaces	*boolean*	Output interfaces.
outputAll	*boolean*	Flood to all interfaces except ingress.
outputController	*boolean*	Send to controller.
outputDrop	*boolean*	Drop packet.
outputFlood	*boolean*	Flood to all non-blocked interfaces except ingress.
outputLocal	*boolean*	Send to switch CPU.
outputNormal	*boolean*	Process normally.
outputQ	*boolean*	Output queue.
[setGroup]	*boolean*	Forward to group.
[ttlDec]	*boolean*	Decrement TTL.
vlan	*boolean*	Set Vlan ID.
vlanPCP	*boolean*	Set Vlan PCP.

2.1136.1.2 OpenFlowTableProfileMatch

ethType	*boolean*	Ethernet type.
icmpCode **	*boolean*	ICMP code. **Since: 4.23.1
icmpType **	*boolean*	ICMP type. **Since: 4.23.1
inInterface	*boolean*	Ingress interface.
ipDst	*boolean*	IPv4 destination address.
ipPortDst	*boolean*	TCP/UDP destination port.
ipPortSrc	*boolean*	TCP/UDP source port.
ipProto	*boolean*	IPv4 protocol.
ipSrc	*boolean*	IPv4 source address.
ipTos	*boolean*	IPv4 TOS.
macDst	*boolean*	Ethernet destination address.
macSrc	*boolean*	Ethernet source address.
vlan	*boolean*	Vlan ID.
vlanPCP	*boolean*	VLAN priority.

2.1137 show openflow queues

controllerQ	*list of OpenFlowQStats* see 2.1137.1	Controller Queues.
portQ	*list of PortQ* see 2.1137.2	Ports.

2.1137.1 OpenFlowQStats

ID	*integer*	Q id.
byteCount	*integer*	Byte count.
errCount	*integer*	Error count.
packetCount	*integer*	Packet count.

2.1137.2 PortQ

interface	*interface name*	Interface.
port	*integer*	Port ID.
stats	*list of OpenFlowQStats* see 2.1137.1	Port Queues.

2.1138 show openflow statistics

activityStats	*list of Stats* see 2.1138.1	Activity stats for 5 second intervals.

2.1138.1 Stats

endTime	*integer*	End time of this interval.
flowEntryCount	*integer*	Number of flows at the end of this interval.

flowModifications	*integer*	Number of messages modified by flows.
packetsDropped	*integer*	Packets dropped instead of being sent to controller.
packetsOut	*integer*	Number of packets.
packetsToController	*integer*	Packets sent to controller.

2.1139 show openstack (ip | ipv6) access-list [...]

see 2.53

2.1140 show openstack agent uuid

uuid	*string*	OpenStack agent UUID.

2.1141 show openstack config all-regions

regions	*table(string) = RegionConfig* see 2.1141.1	Collection of regions.

2.1141.1 RegionConfig

regionName	*string*	Region Name.
tenants	*table(string) = TenantConfig* see 2.1141.1.1	Collection of tenants within the region.

2.1141.1.1 TenantConfig

[tenantBaremetalInstances] **	*table(string) = BaremetalInstanceConfig* see 2.1141.1.1.1	Collection of baremetal instances for a tenant. **Since: 4.18.1
tenantId	*string*	Tenant Id.
tenantNetworks	*table(string) = NetworkConfig* see 2.1141.1.1.2	Collection of tenant networks for a tenant.
[tenantRouterInstances] **	*table(string) = RouterInstanceConfig* see 2.1141.1.1.3	Collection of Routers for a tenant. **Since: 4.18.1
tenantVmInstances	*table(string) = VmInstanceConfig* see 2.1141.1.1.4	Collection of VMs for a tenant.

2.1141.1.1.1 BaremetalInstanceConfig

baremetalHostId	*string*	Ironic node id on which Baremetal Instance executes.
baremetalInstanceId	*string*	Baremetal Instance Id.
baremetalPorts	*table(string) = PortConfig* see 2.1141.1.1.1.1	Collection of ports for a baremetal instance.

2.1141.1.1.1.1 PortConfig

[hosts] **	*list of string*	Hosts on which the port exists (used to provision vms and routers). **Since: 4.18.1
networkId	*string*	Tenant Network Id.
portId	*string*	Port Id.
[portVlanType] **	*enum: native | access | allowed*	Port Vlan Type. **Since: 4.18.1
segments **	*list of SegmentConfig* see 2.1141.1.1.1.2	Collection of segments in the network. **Since: 4.21.0
[switches] **	*table(string) = SwitchportList* see 2.1141.1.1.1.3	Collection of physical switchports for this logical port keyed by switch id (used to provision baremetal nodes). **Since: 4.18.1

2.1141.1.1.1.2 SegmentConfig

networkId	*string*	Network id associated with the segment.
segmentId	*string*	Segment id.
segmentationType	*enum: vxlan | vlan*	Segmentation Type.
segmentationTypeId	*integer*	VLAN/VxLAN ID.

2.1141.1.1.1.3 SwitchportList

switchports	*list of string*	Connected physical interfaces.

2.1141.1.1.2 NetworkConfig

[dynamicSegment] **	*list of SegmentConfig* see 2.1141.1.1.1.2	Collection of dynamic segments in the network. **Since: 4.18.1
networkId	*string*	Tenant Network Id.
networkName	*string*	Tenant network name.
segmentationType	*enum: vxlan | vlan*	Segmentation type of the first static segment.
segmentationTypeId	*integer*	VLAN/VxLAN ID of the first static segment.
shared	*boolean*	Indicates whether the network is a shared network or not.
staticSegment **	*list of SegmentConfig* see 2.1141.1.1.1.2	Collection of static segments in the network. **Since: 4.18.1

2.1141.1.1.3 RouterInstanceConfig

routerHostId	*string*	Host on which Router Instance executes.
routerInstanceId	*string*	Router Instance Id.
routerPorts	*table(string) = PortConfig* see 2.1141.1.1.1.1	Collection of ports for a Router.

2.1141.1.1.4 VmInstanceConfig

vmHostId	*string*	Host on which VM Instance executes.
vmInstanceId	*string*	VM Instance Id.
vmPorts	*table(string) = PortConfig* see 2.1141.1.1.1.1	Collection of ports for a VM.

2.1142 show openstack config networks vni mapping

vlanToVniMap	*table(string) = VlanToVniMap* see 2.1142.1	VLAN to VNI mapping keyed by OpenStack region names.

2.1142.1 VlanToVniMap

mapping	*table(string) = string*	Mapping from vlan range to vni range.

2.1143 show openstack config region REGION_NAME

tenants	*table(string) = TenantConfig* see 2.1141.1.1	Collection of tenants within the specified region.

2.1144 show openstack config region REGION_NAME timestamp

regionName	*string*	Region name.
regionTimestamp	*string*	OpenStack agent UUID.

2.1145 show openstack features

features	*table(string) = integer*	OpenStack Features.

2.1146 show openstack grace-period

gracePeriod	*floating point number*	Grace period in seconds for which the OpenStack agent waits for OpenStack region data.

2.1147 show openstack instances [...]

Could also be an instance of:
Vms 2.1147.1.2.4

[detail]	*boolean*	Display detailed information.
[hosts]	*table(string) = Host* see 2.1122.1	Collection of hosts in topology.
[neighbors]	*table(string) = DirectedEdge* see 2.1123.1	Collection of neighbors in topology.
regions	*table(string) = Region* see 2.1147.1	Collection of regions.

2.1147.1 Region

regionName	*string*	Region Name.
[serviceEndPoints]	*table(string) = ServiceEndPoint* see 2.1147.1.1	Collection of serviceendpoints within the region.
[syncInterval]	*floating point number*	ML2 Sync Interval.
[syncStatus]	*enum: syncInProgress | syncTimedout | unknown | syncComplete*	Sync Status.
tenants	*table(string) = Tenant* see 2.1147.1.2	Collection of tenants within the region.

2.1147.1.1 ServiceEndPoint

authUrl	*string*	Service Auth URL.
name	*string*	Service Name.
[tenantname]	*string*	Keystone admin tenant name.
[username]	*string*	Keystone admin user name.

2.1147.1.2 Tenant

[tenantBaremetalInstances] **	*table(string) = BaremetalInstance* see 2.1147.1.2.1	Collection of baremetal instances for a tenant. **Since: 4.18.1
tenantId	*string*	Tenant Id.
[tenantName]	*string*	Tenant Name.
tenantNetworks	*table(string) = Network* see 2.1147.1.2.2	Collection of tenant networks for a tenant.
tenantRouterInstances **	*table(string) = RouterInstance* see 2.1147.1.2.3	Collection of routers for a tenant. **Since: 4.18.1
tenantVmInstances	*table(string) = VmInstance* see 2.1147.1.2.4	Collection of VMs for a tenant.

2.1147.1.2.1 BaremetalInstance

baremetalHostId	*string*	Ironic node id on which Baremetal Instance executes.

baremetalInstanceId	*string*	Baremetal Instance Id.
[baremetalInstanceName]	*string*	Baremetal Instance Name.
baremetalPorts	*table(string) = Port* see 2.1147.1.2.1.1	Collection of ports for a Baremetal instance.

2.1147.1.2.1.1 Port

[hosts] **	*list of string*	Hosts on which the port exists (used to provision vms and routers). **Since: 4.18.1
networkId	*string*	Tenant Network Id.
portId	*string*	Port Id.
[portName]	*string*	Port Name.
[portVlanType] **	*enum: native | access | allowed*	Port Vlan Type. **Since: 4.18.1
segments **	*list of SegmentConfig* see 2.1141.1.1.1.2	List of segments the port is bound to. **Since: 4.21.3
[switches] **	*table(string) = SwitchportList* see 2.1141.1.1.1.3	Collection of physical switchports for this logical port keyed by switch id (used to provision baremetal nodes). **Since: 4.18.1

2.1147.1.2.2 Network

[dhcpInstances]	*table(string) = DhcpInstance* see 2.1147.1.2.2.2	Collection of DHCP instances.
mappedVni	*integer*	VNI to which the VLAN maps to.
[networkBaremetalInstancesIdName] **	*table(string) = BaremetalInstanceIdName* see 2.1147.1.2.2.1	Collection of baremetal instances for a tenant. **Since: 4.18.1
networkId	*string*	Tenant Network Id.
[networkName]	*string*	Tenant Network Name.
[networkVmInstancesIdName]	*table(string) = VmInstanceIdName* see 2.1147.1.2.2.3	Collection of VMs for a tenant.
segmentationType	*enum: vxlan | vlan*	Segmentation Type.
segmentationTypeId	*integer*	VLAN/VxLAN ID.
shared	*boolean*	Indicates whether the network is a shared network or not.

2.1147.1.2.2.1 BaremetalInstanceIdName

baremetalHostId	*string*	Ironic node id on which Baremetal Instance executes.
baremetalInstanceId	*string*	Baremetal Instance Id.
[baremetalInstanceName]	*string*	Baremetal Instance Name.
[switches]	*table(string) = SwitchportList* see 2.1141.1.1.1.3	Collection of physical switchports for this baremetal on a single network keyed by switch id.

2.1147.1.2.2.2 DhcpInstance

dhcpHostId	*string*	Host on which DHCP Instance executes.
dhcpInstanceId	*string*	DHCP Instance Id.
dhcpPorts	*table(string) = Port* see 2.1147.1.2.1.1	Collection of ports for a DHCP instance.

2.1147.1.2.2.3 VmInstanceIdName

[switches] **	*table(string) = SwitchportList* see 2.1141.1.1.1.3	Collection of physical switchports for this vm on a single network keyed by switch id. **Since: 4.18.1
vmHostId	*string*	Host on which VM Instance executes.
vmInstanceId	*string*	VM Instance Id.
[vmInstanceName]	*string*	VM Instance Name.

2.1147.1.2.3 RouterInstance

routerHostId	*string*	Host on which the router in placed on.
routerInstanceId	*string*	Router Instance Id.
routerInstanceName	*string*	Router Instance Name.
routerPorts	*table(string) = Port* see 2.1147.1.2.1.1	Collection of ports for a router instance.

2.1147.1.2.4 VmInstance

vmHostId	*string*	Host on which VM Instance executes.
vmInstanceId	*string*	VM Instance Id.
[vmInstanceName]	*string*	VM Instance Name.
vmPorts	*table(string) = Port* see 2.1147.1.2.1.1	Collection of ports for a VM.
[detail]	*boolean*	Display detailed information.
[hosts]	*table(string) = Host* see 2.1122.1	Collection of hosts in topology.
[neighbors]	*table(string) = DirectedEdge* see 2.1123.1	Collection of neighbors in topology.
regions	*table(string) = Region* see 2.1147.1	Collection of regions.

2.1148 show openstack networks [...]

[detail]	*boolean*	Display detailed information.
[hosts]	*table(string) = Host* see 2.1122.1	Collection of hosts in topology.
[neighbors]	*table(string) = DirectedEdge* see 2.1123.1	Collection of neighbors in topology.

regions	*table(string) = Region* see 2.1147.1	Collection of regions.

2.1149 show openstack physical-network [...]

[detail]	*boolean*	Display detailed information.
[hosts]	*table(string) = Host* see 2.1122.1	Collection of hosts in topology.
[neighbors]	*table(string) = DirectedEdge* see 2.1123.1	Collection of neighbors in topology.
regions	*table(string) = Region* see 2.1147.1	Collection of regions.
switches	*table(string) = Switch* see 2.1149.1	Collection of switches.

2.1149.1 Switch

hostname	*string*	Hostname configured by user.
interfaces	*table(string) = Interface* see 2.1149.1.1	Collection of interfaces on this switch.
switchId	*string*	Unique identifier (typically MAC-addr) for the switch.

2.1149.1.1 Interface

allowedVlans	*string*	Allowed list of trunk vlans on the interface.
name	*string*	Interface name (Ethernet1, Management1 etc).
[nativeVlan] **	*integer*	Native vlan on the interface. **Since: 4.18.1

2.1150 show openstack regions [...]

regions	*table(string) = Region* see 2.1147.1	Collection of regions.

2.1151 show openstack resource-pool vlan region REGION_NAME uuid

uuid	*string*	OpenStack VLAN assignment UUID.

2.1152 show openstack resource-pools [...]

physicalNetwork	*table(string) = PhysicalNetworkVlanPool* see 2.1152.1	Physical networks keyed by OpenStack region names.

2.1152.1 PhysicalNetworkVlanPool

vlanPool	*table(string) = VlanPool* see 2.1152.1.1	VLAN pools keyed by physical network names.

2.1152.1.1 VlanPool

allocatedVlans	*string*	VLANs that are being used by the OpenStack region.
assignedVlans	*string*	VLANs assigned to the OpenStack region.
availableVlans	*string*	VLANs that are unused in the OpenStack region'sVLAN assignment.

2.1153 show openstack segments { (host HOSTNAME) | (network NETWORK_NAME_OR_ID) | (port PORT_NAME_OR_ID) | (instance INSTANCE_NAME_OR_ID) | (region REGION_NAME) | (switch SWITCH) | (tenant TENANT_NAME_OR_ID) }

regions	*table(string) = RegionInstanceBinding* see 2.1153.1	Collection of instance in region.

2.1153.1 RegionInstanceBinding

instances	*table(string) = InstancePortBinding* see 2.1153.1.1	Collection of ports keyed by instance Id.

2.1153.1.1 InstancePortBinding

instanceId	*string*	Instance Id.
instanceName	*string*	Instance Name.
ports	*table(string) = PortBinding* see 2.1153.1.1.1	Collection of ports in instance keyed by portId.
tenantId	*string*	Tenant Id.
tenantName	*string*	Tenant name.

2.1153.1.1.1 PortBinding

[hosts]	*table(string) = NetworkSegment* see 2.1153.1.1.1.1	Hosts on which the port exists keyed by host name.
portId	*string*	Port Id.
[switches]	*table(string) = NetworkSegment* see 2.1153.1.1.1.1	Physical switches on which the port is connected to keyed by switch name and interface.

2.1153.1.1.1.1 NetworkSegment

networkId	*string*	Tenant Network Id.

networkName	*string*	Tenant network name.
segments	*list of SegmentConfig* see 2.1141.1.1.1.2	List of segments on the network.

2.1154 show openstack sync-timeout

regionSyncTimeout	*floating point number*	Timeout value in seconds for region syncronization.

2.1155 show openstack tenants [...]

regions	*table(string) = Region* see 2.1147.1	Collection of regions.

2.1156 show openstack vlans

switches	*table(string) = Switch* see 2.1149.1	Collection of switches and their vlan information.

2.1157 show openstack vms [...]

see 2.1147.1.2.4

2.1158 show ospfv3 [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfShowInstModel* see 2.1158.1	OSPFv3 per AF instance information for all VRFs.

2.1158.1 AddressFamilyOspf3MultiAfShowInstModel

addressFamily	*table(string) = Ospf3MultiAfShowInstModel* see 2.1158.1.1	OSPFv3 instance information indexed by address family.

2.1158.1.1 Ospf3MultiAfShowInstModel

abrInstance	*boolean*	This OSPFv3 instance is an Area Border Router.
[adjacencyExchangeStart-Threshold]	*integer*	Adjacency exchange-start threshold.
[areaList]	*table(IPv4 address) = Ospf3MultiAfShowInstArea-Model* see 2.1158.1.1.1	List of areas in this OSPFv3 instance.
asbrInstance	*boolean*	This OSPFv3 instance is an AS Border Router.
[ecmpMaximumNexthops]	*integer*	Maximum number of ECMP nexthops.
fipsStatus **	*boolean*	Indicates if FIPS mode is successfully enabled. **Since: 4.20.5
floodPacing **	*integer*	Flood packet pacing timer in msecs. **Since: 4.20.0
[gracefulRestartInfo] **	*OspfShowInstGraceful-RestartInfoModel* see 2.642.1.1.4	Graceful restart information. **Since: 4.18.0
[instanceId]	*integer*	OSPFv3 instance ID.
[lsaInformation]	*Ospf3ShowInstLsaThrottle-TimersInfoModel* see 2.749.1.1.3	LSA information.
lsaLimitInformation **	*Ospf3ShowInstLsaLimit-InfoModel* see 2.749.1.1.2	LSA limit information. **Since: 4.26.2
[maxMetricTypeInformation]	*Ospf3ShowInstMaxMetric-TypeInfoModel* see 2.749.1.1.4	Max metric type information.
[maxMetricUnsetInformation]	*Ospf3ShowInstMaxMetric-UnsetInfoModel* see 2.749.1.1.5	Max metric unset information.
numAreas	*integer*	Number of areas.
numBackboneNeighbors **	*integer*	Number of backbone neighbors. **Since: 4.20.5
numNeighbors	*integer*	Number of neighbors.
numNormalAreas	*integer*	Number of normal areas.
numNssaAreas	*integer*	Number of nssa areas.

numStubAreas	*integer*	Number of stub areas.
[referenceBandwidth]	*integer*	Auto-cost reference bandwidth in Mbps.
[routerId]	*IPv4 address*	OSPFv3 Router id.
shutDown	*boolean*	Indicates if OSPFv3 instance is currently shut down.
[spfInformation]	*Ospf3ShowInstSpfThrottle-TimersInfoModel* see 2.749.1.1.6	SPF information.
[vrf]	*string*	VRF name.

2.1158.1.1.1 Ospf3MultiAfShowInstAreaModel

[areaAuthTypeInformation]	*Ospf3AreaAuthInfoModel* see 2.749.1.1.1.1	Area authentication information.
[areaFilterPrefixList] **	*string*	Area Filter prefix-list. **Since: 4.26.2
areaFiltersConfigured **	*boolean*	Area summary filters configured. **Since: 4.26.0
areaId	*IPv4 address*	OSPFv3 area ID.
areaType	*enum: stubArea | nssaArea | normalArea*	Type of area(normal/stub/nssa).
nssaTransState	*boolean*	Nssa translate always flag for area.
numIntf	*integer*	Number of interfaces in the area.
rangeList	*table(IPv4 or IPv6 address) = Ospf3MultiAfShowInst-AreaRangeModel* see 2.1158.1.1.1.1	List of address ranges in this OSPFv3 area.
spfCount **	*integer*	Area SPF count. **Since: 4.20.5

2.1158.1.1.1.1 Ospf3MultiAfShowInstAreaRangeModel

address	*IP address*	Area range address.
cost	*integer*	Area range cost.
[doNotAdvertise]	*boolean*	Do not advertise the area range. The cost value is invalid if this flag is set to true.
maskLen	*integer*	Area range mask length.

2.1159 show ospfv3 access-list [...]

see 2.53

2.1160 show ospfv3 border-routers [...]

vrfs	*table(string) = AddressFamilyOspf3Multi-AfBorderRoutersInstance* see 2.1160.1	OSPFv3 per AF border routers information for all VRFs.

2.1160.1 AddressFamilyOspf3MultiAfBorderRoutersInstance

addressFamily	*table(string) = Ospf3MultiAfBorderRoutersInstance* see 2.1160.1.1	OSPFv3 instance information indexed by address family.

2.1160.1.1 Ospf3MultiAfBorderRoutersInstance

instanceId	*integer*	OSPFv3 instance identifier.
ospf3BorderRouterEntries	*table(IPv4 address) = Ospf3BorderRoutersEntry* see 2.751.1.1.1	Dictionary of OSPFv3 border router entries indexed by area-id.
vrf	*string*	VRF name.

2.1161 show ospfv3 counters [...]

vrfs	*table(string) = AddressFamilyOspf3-CountersInstance* see 2.1161.1	OSPFv3 Counters.

2.1161.1 AddressFamilyOspf3CountersInstance

addressFamily	*table(string) = Ospf3CountersInstance* see 2.1161.1.1	OSPFv3 instance information indexed by address family.

2.1161.1.1 Ospf3CountersInstance

[lastResetTime]	*floating point number*	UTC timestamp of last counters reset.
lsaTypeCounters	*table(string) = Ospf3CountersLsaEntry* see 2.1161.1.1.1	Dictionary of LSA type counters keyed by counter name.
selfOriginatedLsaReceived	*integer*	Number of self-originated LSAs received.
selfOriginatedLsaReceived-Deleted	*integer*	Number of self-originated LSAs received and deleted.
selfOriginatedLsaReceived-Updated	*integer*	Number of self-originated LSAs received and updated.
spfCount	*integer*	Number of SPF runs.

2.1161.1.1.1 Ospf3CountersLsaEntry

numAsex	*integer*	AS-External LSAs.
numGrc	*integer*	Grace LSAs.
numIap	*integer*	Inter-Area Prefix LSAs.
numIar	*integer*	Inter-Area Router LSAs.
numLink	*integer*	Link LSAs.
numNap	*integer*	Intra-Area Prefix LSAs.
numNetwork	*integer*	Network LSAs.
numNssa	*integer*	NSSA LSAs.
numRouter	*integer*	Router LSAs.

2.1162 show ospfv3 counters interface [...]

vrfs	*table(string) = AddressFamilyOspf3-CountersInterface* see 2.1162.1	OSPFv3 Interface Counters.

2.1162.1 AddressFamilyOspf3CountersInterface

addressFamily	*table(string) = Ospf3CountersInterface* see 2.1162.1.1	OSPFv3 instance information indexed by address family.

2.1162.1.1 Ospf3CountersInterface

interfaces	*table(Interface name)* = *Ospf3CountersInterface-Entry* see 2.1162.1.1.1	Dictionary of OSPFv3 interfaces indexed by the interface name.

2.1162.1.1.1 Ospf3CountersInterfaceEntry

delayedLsAcksSent	*integer*	Number of delayed LS acknowledgements sent.
[lastResetTime]	*floating point number*	UTC timestamp of the last time counters were reset.
lsAcksReceived	*integer*	Number of LS acknowledgements received.
lsUpdatesInFloodQueue	*integer*	Number of LS updates currently in the flood queue.
multicastLsUpdatesSent	*integer*	Number of multicast LS updates sent.

2.1163 show ospfv3 counters neighbor [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfCountersNeighor* see 2.1163.1	OSPFv3 Neighbor Counters.

2.1163.1 AddressFamilyOspf3MultiAfCountersNeighor

addressFamily	*table(string) = Ospf3MultiAfCountersNeighor* see 2.1163.1.1	OSPFv3 instance information indexed by address family.

2.1163.1.1 Ospf3MultiAfCountersNeighor

instanceId	*integer*	Instance ID.
neighborCounterEntries	*list of Ospf3CountersNeighborEntry* see 2.1163.1.1.1	List of neighbor-level counter entries.
vrf	*string*	VRF name.

2.1163.1.1.1 Ospf3CountersNeighborEntry

databaseDescriptionReceived	*integer*	Number of database description packets received.
databaseDescriptionSent	*integer*	Number of database description packets sent.
floodScopeMismatch	*integer*	Number of dropped LSAs due to an AS-external type LSA being received in a stub area.
interfaceName	*interface name*	Name of interface connected to neighbor.
invalidChecksum	*integer*	Number of dropped LSAs due to invalid checksum.
[lastResetTime]	*floating point number*	UTC timestamp of the last time counters were reset.
lsAcknowledgementSent	*integer*	Number of LS acknowledgement packets sent.
lsRequestReceived	*integer*	Number of LS request packets received.
lsRequestSentDueTo-Retransmission	*integer*	Number of LS request packets sent due to retransmission.
lsRequestsInRequestTree	*integer*	Number of LS requests in request tree.
lsUpdateSentDueToLsRequest	*integer*	Number of LS update packets sent due to receiving LS requests.
lsUpdateSentDueToOldLsa-Received	*integer*	Number of LS update packets sent due to receiving old LSAs.
lsUpdateSentDueTo-Retransmission	*integer*	Number of LS update packets sent due to retransmission.
lsUpdatesInRetransmission-List	*integer*	Number of LS updates in retransmission list.
multicastLsAcknowledgement-Received	*integer*	Number of multicast LS acknowledgement packets received.

multicastLsUpdateReceived	*integer*	Number of multicast LS update packets received.
neighborLinkLocalAddress	*IP address*	OSPFv3 neighbor link local address.
neighborRouterId	*IPv4 address*	OSPFv3 neighbor router identifier.
neighborStateChanges	*integer*	Number of neighbor state changes.
newCopyTooQuick	*integer*	Number of dropped LSAs due to LSA arriving before its scheduled next receive time.
newLsRequestSent	*integer*	Number of new LS request packets sent.
prefixLenBig	*integer*	Number of dropped LSAs due to the prefix length of the received LSA being greater than AF mask.
prefixLenInvalid	*integer*	Number of dropped LSAs due to the LSA received whose length of the prefix header was less than (packet's prefix length + length of empty prefix packet).
prefixLenSmall	*integer*	Number of dropped LSAs due to the prefix length of received LSA being less than minimum prefix length.
prefixScopeMismatch	*integer*	Number of dropped LSAs due to the prefix address scope of received LSA not sitelocal or global (only for IPv6 address family).
rtrIdZero	*integer*	Number of dropped LSAs due to the advertising router id of recieved LSA being zero.
type7InNonNssa	*integer*	Number of dropped LSAs due to an NSSA type LSA being received in a non-NSSA area.
unicastLsAcknowledgement-Received	*integer*	Number of unicast LS acknowledgment packets received.
unicastLsUpdateReceived	*integer*	Number of unicast LS update packets received.
unknownLsType	*integer*	Number of dropped LSAs due to an unknown LSA with U-bit set being received in a stub area.
unknownLsaReceived	*integer*	Number of unknown LSAs received.

2.1164 show ospfv3 database [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfDatabaseInstance* see 2.1164.1	OSPFv3 per AF database information for all VRFs.

2.1164.1 AddressFamilyOspf3MultiAfDatabaseInstance

addressFamily	*table(string) = Ospf3MultiAfDatabaseInstance* see 2.1164.1.1	OSPFv3 instance information indexed by address family.

2.1164.1.1 Ospf3MultiAfDatabaseInstance

instanceId	*integer*	OSPFv3 instance ID.
ospf3AreaEntries	*table(IPv4 address) = Ospf3AreaModel* see 2.752.1.1.1	A mapping between an Area and its LSA list.
ospf3AsLsas	*list of Ospf3Lsa* see 2.752.1.1.1.1	List of LSAs with AS scope.
ospf3InterfaceEntries	*table(Interface name) = Ospf3InterfaceModel* see 2.752.1.1.2	A mapping between an Interface and its LSA list.
shutdown **	*boolean*	OSPFv3 instance is currently shut down. **Since: 4.23.0

2.1165 show ospfv3 database database-summary [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfDatabaseSummary* see 2.1165.1	OSPFv3 per AF database summary information for all VRFs.

2.1165.1 AddressFamilyOspf3MultiAfDatabaseSummary

addressFamily	*table(string) = Ospf3MultiAfDatabaseSummary* see 2.1165.1.1	OSPFv3 instance information indexed by address family.

2.1165.1.1 Ospf3MultiAfDatabaseSummary

areaId	*IP address*	Area identifier.
instanceId	*integer*	Instance ID.
numAsex	*integer*	Number of AS-external LSAs.
numGrace	*integer*	Number of Grace LSAs.
numInterPfx	*integer*	Number of Inter Area Prefix LSAs.
numInterRtr	*integer*	Number of Inter Area Router LSAs.
numIntraPfx	*integer*	Number of Intra-Area Prefix LSAs.
numLink	*integer*	Number of Link LSAs.
numNetwork	*integer*	Number of Network LSAs.
numNssa	*integer*	Number of type-7 NSSA LSAs.
numRouter	*integer*	Number of Router LSAs.
vrf	*string*	VRF name.

2.1166 show ospfv3 interface [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfInterface* see 2.1166.1	OSPFv3 per AF interface information for all VRFs.

2.1166.1 AddressFamilyOspf3MultiAfInterface

addressFamily	*table(string) = Ospf3MultiAfInterface* see 2.1166.1.1	OSPFv3 instance information indexed by address family.

2.1166.1.1 Ospf3MultiAfInterface

interfaces	*table(Interface name) = Ospf3InterfaceEntry* see 2.754.1.1.1	Dictionary of OSPFv3 interfaces indexed by the interface name.

2.1167 show ospfv3 lsa-log [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfLsaLogInst* see 2.1167.1	OSPFv3 per AF Lsa Log for all VRFs.

2.1167.1 AddressFamilyOspf3MultiAfLsaLogInst

addressFamily	*table(string) = Ospf3MultiAfLsaLogInst* see 2.1167.1.1	OSPFv3 instance information indexed by address family.

2.1167.1.1 Ospf3MultiAfLsaLogInst

instanceId	*integer*	Private attribute for instance identifier.
ospf3LsaLogEntries	*list of Ospf3LsaLogEntry* see 2.755.1.1.1	List of LSA log entries.
[routerId]	*IPv4 address*	OSPFv3 router identifier.
shutdown **	*boolean*	OSPFv3 instance is currently shut down. **Since: 4.23.0

2.1168 show ospfv3 neighbor [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfNeighborsInstance* see 2.1168.1	OSPFv3 per AF neighbor information for all VRFs.

2.1168.1 AddressFamilyOspf3MultiAfNeighborsInstance

addressFamily	*table(string) = Ospf3MultiAfNeighborsInstance* see 2.1168.1.1	OSPFv3 instance information indexed by address family.

2.1168.1.1 Ospf3MultiAfNeighborsInstance

instanceId	*integer*	OSPFv3 instance ID.
ospf3NeighborEntries	*list of Ospf3NeighborsEntry* see 2.756.1.1.1	List of OSPFv3 neighbor entries.

2.1169 show ospfv3 request-list [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfLsaRequestList* see 2.1169.1	OSPFv3 AF LSA request list information for all VRFs.

2.1169.1 AddressFamilyOspf3MultiAfLsaRequestList

addressFamily	*table(string) = Ospf3MultiAfLsaRequestList* see 2.1169.1.1	OSPFv3 instance information indexed by address family.

2.1169.1.1 Ospf3MultiAfLsaRequestList

instanceId	*integer*	OSPFv3 instance identifier.
neighbors	*table(IPv4 address) = Ospf3LsaRequestInstance* see 2.757.1.1.1	Dictionary of request keyed by neighbor router ID.

2.1170 show ospfv3 retransmission-list [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfLsaRetransmissionList* see 2.1170.1	OSPFv3 per AF LSA retransmission information for all VRFs.

2.1170.1 AddressFamilyOspf3MultiAfLsaRetransmissionList

addressFamily	*table(string) = Ospf3MultiAfLsaRetransmissionList* see 2.1170.1.1	OSPFv3 instance information indexed by address family.

2.1170.1.1 Ospf3MultiAfLsaRetransmissionList

instanceId	*integer*	OSPFv3 instance identifier.
neighbors	*table(IPv4 address) = Ospf3LsaRetransmissionInstance* see 2.758.1.1.1	Dictionary of retransmission list keyed by neighbor router ID.

2.1171 show ospfv3 spf-log [...]

vrfs	*table(string) = AddressFamilyOspf3MultiAfSpfLogInst* see 2.1171.1	OSPFv3 per AF SPF Log for all VRFs.

2.1171.1 AddressFamilyOspf3MultiAfSpfLogInst

addressFamily	*table(string) = Ospf3MultiAfSpfLogInst* see 2.1171.1.1	OSPFv3 instance information indexed by address family.

2.1171.1.1 Ospf3MultiAfSpfLogInst

instanceId	*integer*	Private attribute for instance identifier.
ospf3SpfLogEntries	*list of Ospf3SpfLogEntry* see 2.759.1.1.1	OSPFv3 SPF Log Entries.
[routerId]	*IPv4 address*	OSPFv3 router identifier.
shutdown **	*boolean*	OSPFv3 instance is currently shut down. **Since: 4.23.0

2.1172 show p4-runtime

[devices]	*table(string) = P4RtDeviceStatusEntry* see 2.1172.1	P4Runtime device status entries keyed by device name.
enabled	*boolean*	P4Runtime server is enabled.
numClients	*integer*	Number of P4Runtime client connections.
[transport]	*P4RtTransport* see 2.1172.2	P4Runtime server transport status.

2.1172.1 P4RtDeviceStatusEntry

deviceId	*integer*	P4Runtime device ID.
deviceName	*string*	Device name.
[primaryElectionId]	*P4RuntimeElectionId* see 2.1172.1.1	Primary election ID.

2.1172.1.1 P4RuntimeElectionId

high	*integer*	High 64 bits of the 128-bit election ID.
low	*integer*	Low 64 bits of the 128-bit election ID.

2.1172.2 P4RtTransport

certUsernameAuthn **	*boolean*	Use the username from the certificate for AAA authentication. **Since: 4.30.1
port	*integer*	The port this server is listening on.
[sslProfile] **	*string*	SSL Profile. **Since: 4.30.1
vrfName	*string*	The VRF this server is listening in.

2.1173 show p4-runtime device

[devices]	*table(string) = P4RtDeviceStatusEntryDetail* see 2.1173.1	P4Runtime device status entries keyed by name.

2.1173.1 P4RtDeviceStatusEntryDetail

deviceId	*integer*	P4Runtime device ID.
deviceName	*string*	Device name.
packetMetadata	*list of P4RtControllerPktMetadataEntry* see 2.1173.1.1	Packet metatadatas received.
primaryElectionId	*P4RuntimeElectionId* see 2.1172.1.1	Primary election ID.
writeRpcReceived	*boolean*	Write RPC request received.

2.1173.1.1 P4RtControllerPktMetadataEntry

bitWidth	*integer*	Bit width of the metadata.
messageType	*string*	Type of Message containing the metadata.
metadataId	*integer*	Metadata ID.
name	*string*	Name of the metadata.

2.1174 show p4-runtime device counters packet-io

devicesCounter	*table(string) = P4RtDeviceCounterEntry* see 2.1174.1	A mapping of P4Runtime devices to its packet IO counters.

2.1174.1 P4RtDeviceCounterEntry

packetIn	*integer*	Number of PacketIn responses sent.
packetOut	*integer*	Number of PacketOut requests received.

2.1175 show patch panel [...]

revision 2

patches	*table(string) = Patch* see 2.1175.1	Maps patch names to their respective configuration and operational state.

2.1175.1 Patch

connectors	*table(string) = Connector* see 2.1175.1.1	Maps connector names to their respective configuration.
[lastStatusChange] **	*floating point number*	Time of last time patch state changed. **Since: 4.27.2
status	*enum: partiallyUp | standby | down | adminDown | unprogrammed | cliConflict | up*	Patch status. Admin down - Patch disabled Down - Patch connectors not ready Cli conflict - Patch configuration is invalid Standby - Patch configured but on standby Unprogrammed - Patch configured but not programmed in hardware Up - Patch configured and installed Partially up - Patch configured but only partially installed.

2.1175.1.1 Connector

[alternatives] **	*table(string) = LdpPseudowireConnectorInfo* see 2.1175.1.1.3	a mapping of pwName to ldpPseudowireInfo. **Since: 4.28.0
[bgpPseudowireConnectorInfo]	*BgpPseudowireConnector* see 2.1175.1.1.1	BGP pseudowire connector information.
connType	*enum: port | tagged | bgp | ldp | mplsStatic*	Connector type. tagged - Local interface delimited by an 802.1Q tag value port - Local interface not delimited by 802.1Q tag values ldp - LDP pseudowire bgp - BGP pseudowire.
[ldpPseudowireConnectorInfo]	*LdpPseudowireConnector* see 2.1175.1.1.2	LDP pseudowire connector information.
[mplsStaticPseudowireConnectorInfo] **	*MplsStaticPseudowireConnector* see 2.1175.1.1.4	MPLS static pseudowire connector information. **Since: 4.30.1
[portConnectorInfo]	*PortConnector* see 2.1175.1.1.5	Port connector information.

status	*enum: cannotAllocateVlanId* | *noLocalMapping* | *encapHwProgrammingFailed* | *waitingForVlanId* | *down* | *mtuMismatch* | *noPeerLabel* | *controlWordMismatch* | *neighborEvpn* | *encapInvalidConnConfig* | *partiallyUp* | *decapHwProgrammingFailed* | *connectorUnprogrammed* | *adminDown* | *decapConnInfoMissing* | *unprogrammed* | *cliConflict* | *cliIncomplete* | *hasTaggedIntf* | *encapConnInfoMissing* | *unknown* | *noTunnelRib* | *standby* | *pwTypeMismatch* | *intfMode* | *pwNotForwardingRemote* | *noAdPerESRoute* | *noLocalEnd* | *noRemoteEnd* | *unsupportedTunnelMode* | *encapHwFecUnprogrammed* | *disabledByPw* | *up* | *noTunnel* | *decapInvalidConnConfig*	Connector status.
[taggedConnectorInfo]	*TaggedConnector* see 2.1175.1.1.6	Tagged connector information.

2.1175.1.1.1 BgpPseudowireConnector

[color] **	*integer*	Pseudowire color. **Since: **currentRelease**
[flowLabelUsed]	*enum: disabled* | *both* | *transmit* | *receive*	Flow label used.
[localConfig]	*BgpPseudowireLocalConfig* see 2.1175.1.1.1.1	Local BGP pseudowire config.
[neighborConfigs]	*table(IPv4 or IPv6 address) = BgpPseudowireNeighborConfig* see 2.1175.1.1.1.2	BGP neighbors' pseudowire configs.
[pwName]	*string*	BGP VPWS pseudowire name.
[qosMapName]	*string*	QoS map name.
[qosMapType]	*enum: cosToTc* | *dscpToTc*	QoS map type.
[vpwsName]	*string*	BGP VPWS instance name.
[vpwsType]	*enum: vlan* | *fxc* | *port*	EVPN VPWS type. port - port-based vlan - VLAN-based fxc - flexible cross connect.

2.1175.1.1.1.1 BgpPseudowireLocalConfig

[controlWord]	*boolean*	Control word used.
[label]	*integer*	Pseudowire Label.
[mtu]	*integer*	Maximum transmission unit.

2.1175.1.1.1.2 BgpPseudowireNeighborConfig

[color] **	*integer*	Pseudowire color. **Since: **currentRelease**
[controlWord]	*boolean*	Control word used.
[label]	*integer*	Pseudowire label.
[mtu]	*integer*	Maximum transmission unit.
[status] **	*enum: mtuMismatch | up | standby*	Neighbor status. **Since: 4.27.0
[tunnelIndex]	*integer*	Tunnel ending at this tunnel endpoint.
[tunnelType]	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.1175.1.1.2 LdpPseudowireConnector

[color]	*integer*	Pseudowire color.
controlWord	*boolean*	Control word used.
[flowLabelUsed]	*enum: disabled | both | transmit | receive*	Flow label used.
[localConfig]	*LdpPseudowirePeerConfig* see 2.1175.1.1.2.3	Local LDP pseudowire config.
[name]	*string*	LDP pseudowire name.
[neighbor]	*IP address*	LDP neighbor.
[neighborConfig]	*LdpPseudowirePeerConfig* see 2.1175.1.1.2.3	LDP neighbor's pseudowire config.
[neighborFaultFlags]	*LdpPseudowireErrorFlags* see 2.1175.1.1.2.2	Neighbor pseudowire error flags.
neighborForwarding	*boolean*	Pseudowire forwarding traffic.
[pseudowireId]	*integer*	Pseudowire ID.

pseudowireType	*enum: pwType5 | pwTypeNone | pwType4*	Pseudowire type.
[qosMapName]	*string*	QoS map name.
[qosMapType]	*enum: cosToTc | dscpToTc*	QoS map type.
[tunnels]	*list of TunnelInfo* see 2.1175.1.1.2.1	List of resolving tunnels for this peer.

2.1175.1.1.2.1 TunnelInfo

[color]	*integer*	Tunnel color.
tunnelIndex	*integer*	Tunnel index.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.1175.1.1.2.2 LdpPseudowireErrorFlags

[etInFault]	*boolean*	Ethernet receive fault.
[etOutFault]	*boolean*	Ethernet transmit fault.
[notForwarding]	*boolean*	Pseudowire not forwarding (other reason).
[tunInFault]	*boolean*	Tunnel receive fault.
[tunOutFault]	*boolean*	Tunnel transmit fault.

2.1175.1.1.2.3 LdpPseudowirePeerConfig

[dot1qTagRequested]	*integer*	802.1Q VLAN requested.
[flowLabelCapability]	*enum: disabled | both | transmit | receive*	Flow label config.
[groupId]	*integer*	Group ID.
[intfDescription]	*string*	Interface description.
[label]	*integer*	Pseudowire Label.
[mtu]	*integer*	Maximum transmission unit.
[vccvCcTypes]	*integer*	Supported VCCV CC types.
[vccvCvTypes]	*integer*	Supported VCCV CV types.

2.1175.1.1.3 LdpPseudowireConnectorInfo

[active]	*boolean*	Current active pseudowire.
[alternate]	*boolean*	Indicate pseudowire is alternate or primary.
[color]	*integer*	Pseudowire color.
[neighbor]	*IP address*	LDP neighbor.
[pseudowireId]	*integer*	Pseudowire ID.
status	*enum: cannotAllocateVlanId* | *noLocalMapping* | *encapHwProgrammingFailed* | *waitingForVlanId* | *down* | *mtuMismatch* | *noPeerLabel* | *controlWordMismatch* | *neighborEvpn* | *encapInvalidConnConfig* | *partiallyUp* | *decapHwProgrammingFailed* | *connectorUnprogrammed* | *adminDown* | *decapConnInfoMissing* | *unprogrammed* | *cliConflict* | *cliIncomplete* | *hasTaggedIntf* | *encapConnInfoMissing* | *unknown* | *noTunnelRib* | *standby* | *pwTypeMismatch* | *intfMode* | *pwNotForwardingRemote* | *noAdPerESRoute* | *noLocalEnd* | *noRemoteEnd* | *unsupportedTunnelMode* | *encapHwFecUnprogrammed* | *disabledByPw* | *up* | *noTunnel* | *decapInvalidConnConfig*	ldp pseudowire connector status.

2.1175.1.1.4 MplsStaticPseudowireConnector

[localLabel]	*integer*	Local MPLS label used.
[name]	*string*	MPLS static pseudowire name.
[neighborLabel]	*integer*	Neighbor MPLS label used.
[transportIntf]	*interface name*	MPLS static transport interface.
[tunnelIntfMode] **	*enum: invalidEncap* | *gueEncap* | *greEncap* | *mplsEncap* | *ipInIpEncap* | *srv6TunnelEncap* | *ipSecEncap* | *vxlanEncap* | *ipSecGreEncap* | *dsfEncap* | *srv6SidEncap*	Transport interface tunnel mode. **Since: **currentRelease**

2.1175.1.1.5 PortConnector

[errorDisableTrigger] **	*enum: suppressedByNoErrorDisable | remoteNotForwarding | suppressedByAlternateConnector | none | remoteFailure*	Configuration which will error disable the interface. **Since: 4.28.2
interface	*interface name*	Local interface.

2.1175.1.1.6 TaggedConnector

dot1qTag	*integer*	802.1Q Tag.
interface	*interface name*	Local interface.

2.1176 show patch panel forwarding

revision 3

labelForwardingInfoTable	*LabelForwardingInfoTable* see 2.1176.1	Label forwarding information table.
portForwardingInfoTable	*PortForwardingInfoTable* see 2.1176.2	Port forwarding information table.
taggedForwardingInfoTable	*TaggedForwardingInfoTable* see 2.1176.3	Tagged forwarding information table.

2.1176.1 LabelForwardingInfoTable

labelForwardingInfos	*table(numeric string) = LabelForwardingInfo* see 2.1176.1.1	a mapping of decap labels to Label forwarding information entries.

2.1176.1.1 LabelForwardingInfo

[flowLabel]	*string*	Flow label used.
label	*integer*	Pseudowire decap label.
outConnInfo	*ForwardingConnectorInfo* see 2.1176.1.1.1	(deprecated; use outConnectors).
outConnectors	*list of ForwardingConnectorInfo* see 2.1176.1.1.1	Out connectors.
patch	*string*	Patch name.
pseudowireType	*enum: pwType5 | pwTypeNone | pwType4*	Pseudowire type.
source	*enum: LDP | EVPN | MPLS static*	Pseudowire source.

status	*enum: cannotAllocateVlanId | noLocalMapping | encapHwProgrammingFailed | waitingForVlanId | down | mtuMismatch | noPeerLabel | controlWordMismatch | neighborEvpn | encapInvalidConnConfig | partiallyUp | decapHwProgrammingFailed | connectorUnprogrammed | adminDown | decapConnInfoMissing | unprogrammed | cliConflict | cliIncomplete | hasTaggedIntf | encapConnInfoMissing | unknown | noTunnelRib | standby | pwTypeMismatch | intfMode | pwNotForwardingRemote | noAdPerESRoute | noLocalEnd | noRemoteEnd | unsupportedTunnelMode | encapHwFecUnprogrammed | disabledByPw | up | noTunnel | decapInvalidConnConfig*	Connector status.

2.1176.1.1.1 ForwardingConnectorInfo

[labelConnFwdingInfo]	*LabelConnectorFwdingInfo* see 2.1176.1.1.1.1	Label connector forwarding information.
[portConnFwdingInfo]	*PortConnectorFwdingInfo* see 2.1176.1.1.1.3	Port connector forwarding information.
[taggedConnFwdingInfo]	*TaggedConnectorFwdingInfo* see 2.1176.1.1.1.4	Tagged connector forwarding information.

2.1176.1.1.1.1 LabelConnectorFwdingInfo

peers	*table(IPv4 or IPv6 address) = LabelConnectorFwdingPeerInfo* see 2.1176.1.1.1.2	List of peers for this connector.

2.1176.1.1.1.2 LabelConnectorFwdingPeerInfo

[encapControlWord]	*boolean*	Control word used to encapsulate packets toward this peer.
peerLabel	*integer*	Label received from the peer.

tunnels	*list of TunnelInfo* see 2.1175.1.1.2.1	List of resolving tunnels for this peer.

2.1176.1.1.1.3 PortConnectorFwdingInfo

[decapControlWord]	*boolean*	Control word used to decapsulate packets toward this AC.
interface	*interface name*	Local interface.

2.1176.1.1.1.4 TaggedConnectorFwdingInfo

[decapControlWord]	*boolean*	Control word used to decapsulate packets toward this AC.
dot1qVlan	*integer*	802.1Q VLAN.
interface	*interface name*	Local interface.

2.1176.2 PortForwardingInfoTable

portForwardingInfos	*table(string) = PortForwardingInfo* see 2.1176.2.1	A mapping of port connectors to connector forwarding information entries.

2.1176.2.1 PortForwardingInfo

[flowLabel]	*string*	Flow label used.
inConnInfo	*ForwardingConnectorInfo* see 2.1176.1.1.1	In connector.
outConnInfo	*ForwardingConnectorInfo* see 2.1176.1.1.1	Out connector.
patch	*string*	Patch name.
[pseudowireType]	*enum: pwType5 | pwTypeNone | pwType4*	Pseudowire type.
[source]	*enum: LDP | EVPN | MPLS static*	Pseudowire source.

status	*enum: cannotAllocateVlanId* | *noLocalMapping* | *encapHwProgrammingFailed* | *waitingForVlanId* | *down* | *mtuMismatch* | *noPeerLabel* | *controlWordMismatch* | *neighborEvpn* | *encapInvalidConnConfig* | *partiallyUp* | *decapHwProgrammingFailed* | *connectorUnprogrammed* | *adminDown* | *decapConnInfoMissing* | *unprogrammed* | *cliConflict* | *cliIncomplete* | *hasTaggedIntf* | *encapConnInfoMissing* | *unknown* | *noTunnelRib* | *standby* | *pwTypeMismatch* | *intfMode* | *pwNotForwardingRemote* | *noAdPerESRoute* | *noLocalEnd* | *noRemoteEnd* | *unsupportedTunnelMode* | *encapHwFecUnprogrammed* | *disabledByPw* | *up* | *noTunnel* | *decapInvalidConnConfig*	Connector status.

2.1176.3 TaggedForwardingInfoTable

taggedForwardingInfos	*table(string) = TaggedForwardingInfo* see 2.1176.3.1	A mapping of tagged connectors to connector forwarding information entries.

2.1176.3.1 TaggedForwardingInfo

[dot1qVlan]	*integer*	802.1Q VLAN.
[flowLabel]	*string*	Flow label used.
inConnInfo	*ForwardingConnectorInfo* see 2.1176.1.1.1	In connector.
outConnInfo	*ForwardingConnectorInfo* see 2.1176.1.1.1	Out connector.
patch	*string*	Patch name.
[pseudowireType]	*enum: pwType5* | *pwTypeNone* | *pwType4*	Pseudowire type.
[source]	*enum: LDP* | *EVPN* | *MPLS static*	Pseudowire source.

status	*enum:* *cannotAllocateVlanId* | *noLocalMapping* | *encapHwProgrammingFailed* | *waitingForVlanId* | *down* | *mtuMismatch* | *noPeerLabel* | *controlWordMismatch* | *neighborEvpn* | *encapInvalidConnConfig* | *partiallyUp* | *decapHwProgrammingFailed* | *connectorUnprogrammed* | *adminDown* | *decapConnInfoMissing* | *unprogrammed* | *cliConflict* | *cliIncomplete* | *hasTaggedIntf* | *encapConnInfoMissing* | *unknown* | *noTunnelRib* | *standby* | *pwTypeMismatch* | *intfMode* | *pwNotForwardingRemote* | *noAdPerESRoute* | *noLocalEnd* | *noRemoteEnd* | *unsupportedTunnelMode* | *encapHwFecUnprogrammed* | *disabledByPw* | *up* | *noTunnel* | *decapInvalidConnConfig*	Connector status.

2.1177 show patch panel summary

bgpVpwsCount	*table(string) = integer*	BGP EVPN VPWS patch count by status.
ldpCount	*table(string) = integer*	LDP patch count by status.
localCount	*table(string) = integer*	Local patch count by status.
mplsStaticCount **	*table(string) = integer*	MPLS static patch count by status. **Since: 4.30.1
patchCount	*integer*	Patch count.

2.1178 show path-selection application counters [...]

lbProfiles	*table(numeric string) = DpsLbProfiles* see 2.1178.1	A mapping of load-balance profile index to its application counters.

2.1178.1 DpsLbProfiles

appProfile	*string*	Name of the application profile.
flows	*integer*	Number of flows for the application.
outBytes	*integer*	Number of out bytes for the application.

outPkts	*integer*	Number of out packets for the application.
[pathGroups]	*table(string) = PathGroups* see 2.1178.1.1	A mapping of group name to its counters.
peerIp	*IP address*	Peer IP address of the ITS session.
peerName **	*string*	Name of the peer. **Since: 4.23.2
throughput	*floating point number*	Throughput in Mbps for the application.
vrf	*string*	Name of the VRF.

2.1178.1.1 PathGroups

[paths]	*table(numeric string) = Paths* see 2.1178.1.1.1	A mapping of path index to its counters.

2.1178.1.1.1 Paths

flows	*integer*	Number of flows going through the path.
outBytes	*integer*	Number of bytes going via the path.
outPkts	*integer*	Number of packets going via the path.
throughput	*floating point number*	Throughput in Mbps of the path.

2.1179 show path-selection load-balance counters [...]

lbProfiles	*table(numeric string) = DpsLbProfiles* see 2.1178.1	A mapping of load-balance profile index to its path selection counters.

2.1180 show path-selection mtu [...]

dpsPeers **	*table(IPv4 or IPv6 address) = DpsPeer* see 2.1180.1	A mapping of peer IP to peer path groups. **Since: 4.23.2

2.1180.1 DpsPeer

[avtRegionId] **	*integer*	Adaptive virtual topology region ID. **Since: 4.29.2
[avtSiteId] **	*integer*	Adaptive virtual topology site ID. **Since: 4.29.2
[avtZoneId] **	*integer*	Adaptive virtual topology zone ID. **Since: 4.29.2
dpsGroups	*table(string) = DpsGroup* see 2.1180.1.1	A mapping of group name to its paths.
peerName	*string*	Peer name.

2.1180.1.1 DpsGroup

dpsPaths	*table(string) = DpsPath* see 2.1180.1.1.1	A mapping of path name to its state.

2.1180.1.1.1 DpsPath

destination	*string*	Destination IP address of the path group.
[destinationPort] **	*integer*	Destination UDP port of the DPS path. **Since: 4.27.1
[destinationWanId] **	*integer*	Destination WAN ID of the DPS path. **Since: 4.27.1
dpsSessions **	*table(numeric string) = DpsSession* see 2.1180.1.1.1.1	A mapping of traffic class to DPS session. **Since: 4.23.2
[label] **	*integer*	Label value of the DPS path. **Since: 4.29.1
[localIntf] **	*interface name*	Local interface name of this path. **Since: 4.29.2
[mtu] **	*integer*	Path MTU. **Since: 4.27.2
[mtuConfigType] **	*enum: disabled | auto | static*	Type of MTU config. **Since: 4.27.2
[mtuDiscDueSecs] **	*integer*	MTU discovery due in secs. **Since: 4.27.2
[mtuDiscIntervalSecs] **	*integer*	MTU discovery interval in secs. **Since: 4.28.0
[mtuDiscTimeTakenMsecs] **	*floating point number*	MTU discovery time taken in msecs. **Since: 4.27.2
[mtuErrStateReason] **	*enum: mtu outside defined range | none | response message timeout | path down event*	MTU error state reason. **Since: 4.28.0

[mtuNumProbesRcvd] **	*integer*	Number of Probes received. **Since: 4.27.2
[mtuNumProbesSent] **	*integer*	Number of Probes sent. **Since: 4.27.2
[mtuNumReportReqRcvd] **	*integer*	Number of Report Requests received. **Since: 4.27.2
[mtuNumReportReqSent] **	*integer*	Number of Report Requests sent. **Since: 4.27.2
[mtuNumReportRespRcvd] **	*integer*	Number of Report Responses received. **Since: 4.27.2
[mtuNumReportRespSent] **	*integer*	Number of Report Responses sent. **Since: 4.27.2
[mtuNumReqProcErr] **	*integer*	Number of Request processing error. **Since: 4.27.2
[mtuNumRespProcErr] **	*integer*	Number of Reponse processing error. **Since: 4.27.2
[mtuState] **	*enum: error* | *done* | *notApplicable* | *init* | *probing*	MTU discovery state. **Since: 4.27.2
pathType **	*enum: dynamic* | *static*	Type of the path. **Since: 4.26.2
source	*string*	Source IP address of the path group.
[sourcePort] **	*integer*	Source UDP port of the DPS path. **Since: 4.27.1
[sourceWanId] **	*integer*	Source WAN ID of the DPS path. **Since: 4.27.1
state	*enum: routePending* | *routeResolved* | *routeStateInvalid* | *ipsecEstablished* | *arpPending* | *ipsecPending*	Routing state of this path.

2.1180.1.1.1.1 DpsSession

active	*boolean*	The telemetry session state is active.
seconds	*integer*	Number of seconds the telemetry session has been in this state.

2.1181 show path-selection paths [...]

see 2.1180

2.1182 show path-selection paths export

[vtep]	*IP address*	VTEP IP.
wans	*table(numeric string) = DpsExport* see 2.1182.1	A mapping of WAN ID to exported endpoint.

2.1182.1 DpsExport

intf	*interface name*	Local interface.
ipAddress	*IP address*	Exported IP address.
ipSource	*enum: stun* | *interface* | *intfpublic*	Source of the IP address.
pathGroup	*string*	Path group name.
port	*integer*	Exported port number.

2.1183 show pci

pciIds	*table(string) = PciErrorSum* see 2.1183.1	Summary of error counters for each PCI device.
switchs	*table(string) = integer*	Summary of smBus errors on PCIe switch.

2.1183.1 PciErrorSum

correctableErrors	*integer*	Correctable error counter.
fatalErrors	*integer*	Uncorrectable fatal error counter.
linkSpeed	*floating point number*	Current link speed of the PCI device in GT/s.
linkWidth	*integer*	Current link width of the PCI device.
name	*string*	PCI device name.
nonFatalErrors	*integer*	Uncorrectable non-fatal error counter.

2.1184 show pci detail

devices	*table(string) = PciDetailError* see 2.1184.1	Summary of pci device errors.

2.1184.1 PciDetailError

name	*string*	Name of pci device.
pciErrors	*table(string) = integer*	Error counts.
severity	*integer*	Uncorrectable errors severity.

2.1185 show pci tree

[devices]	*table(string) = PciTreeNode* see 2.1185	PCI children of this device

2.1186 show peer-filter [...]

peerFilters	*table(string) = PeerFilter* see 2.1186.1	Peer filters by name.

2.1186.1 PeerFilter

description	*list of string*	Peer filter description.
entries	*table(numeric string) = PeerFilterEntry* see 2.1186.1.1	Peer filter entries by sequence number.

2.1186.1.1 PeerFilterEntry

filterType	*enum: reject | accept*	Filter type.
matchRules	*PeerFilterMatchRules* see 2.1186.1.1.1	Match rules.

2.1186.1.1.1 PeerFilterMatchRules

[asRange]	*AsnRange* see 2.1186.1.1.1.1	BGP ASN range.

2.1186.1.1.1.1 AsnRange

[end]	*integer*	ASN range end.
start	*integer*	ASN range start.

2.1187 show pim (ipv4 | ipv6) anycast-rp [...]

see 2.662

2.1188 show pim (ipv4 | ipv6) bsr [...]

see 2.667

2.1189 show pim (ipv4 | ipv6) bsr counters [...]

see 2.668

2.1190 show pim (ipv4 | ipv6) bsr debug

see 2.669

2.1191 show pim (ipv4 | ipv6) bsr rp [...]

see 2.23

2.1192 show pim (ipv4 | ipv6) bsr rp-hash GROUP [...]

see 2.24

2.1193 show pim (ipv4 | ipv6) config-sanity anycast-rp [...]

downIntfs	*list of Interface name*	Anycast RP interfaces which are down.
notAnycastRPs	*list of IP address*	This RP address is not configured as anycast RP.
notCfgRPs	*list of IP address*	Anycast RPs not configured on the router.
notIntfCfgRPs	*list of IP address*	Anycast RP not configured on any interface.
notLoopback	*list of Interface name*	Anycast RP interfaces which are not loopback.
notRPs	*list of IP address*	Anycast RPs not configured as RP addresses.
routeCheck	*list of GenRouteCheck* see 2.1193.1	route check hints.

2.1193.1 GenRouteCheck

hintsFound	*integer*	number of hints found.
noRoute	*boolean*	No route to the prefix.
nodeType	*string*	Node type.
nonPimHop	*list of IP address*	Route via non-Pim hops.
nonPimIntf	*list of Interface name*	Route via non-Pim interface.
prefix	*IP address*	Route check for this prefix.

2.1194 show pim (ipv4 | ipv6) interface [...]

see 2.678

2.1195 show pim (ipv4 | ipv6) neighbor [...]

revision 2

see 2.679

2.1196 show pim (ipv4 | ipv6) neighbor bfd [...]

revision 2

see 2.680

2.1197 show pim (ipv4 | ipv6) register-source [...]

see 2.682

2.1198 show pim (ipv4 | ipv6) rp [...]

revision 2

see 2.683

2.1199 show pim (ipv4 | ipv6) rp-candidate [...]

see 2.684

2.1200 show pim evpn gateway dr [...]

`vlans`	*table(numeric string) = EvpnGatewayDr* see 2.1200.1	Mapping of VLAN to external gateway DR information.

2.1200.1 EvpnGatewayDr

`addr`	*IP address*	Address of Designated Router.
`[algorithm]` **	*enum: preference | hrw | modulus*	Algorithm used to elect DR. **Since: 4.27.1
`role`	*enum: dr | nonDr*	Router role on this VLAN.

2.1201 show pim ipv4 sparse-mode non-dr drop-rules [...]

`interfaces`	*table(Interface name) = NonDrDropRuleInterface* see 2.1201.1	Map of interfaces for non DR install oifs feature.

2.1201.1 NonDrDropRuleInterface

[inactiveReason]	*enum: Drop rule programming failed on interface* | *Interface is DR* | *Drop rule programming timed out* | *Drop rule unprogramming timed out* | *Conflicting drop rule programmed on interface* | *Multicast routes forwarding* | *Drop rule creation failed* | *Waiting for platform to program drop rule* | *Drop rule unprogramming failed on interface*	Reason why state is inactive or error.
numRoutesForwarding	*integer*	Number of multicast routes forwarding on this interface.
numRoutesReadyToForward **	*integer*	Number of multicast routes ready to forward on this interface. **Since: 4.24.0
state	*enum: unprogramming* | *error* | *inactive* | *programming* | *active* | *init* | *unprogrammed*	Non-DR drop rule state for interface.

2.1202 show pim ipv6 sparse-mode join [...]

see 2.25

2.1203 show pim ipv6 sparse-mode route IP1 IP2 interface [...]

see 2.20

2.1204 show pim ipv6 sparse-mode route [...]

see 2.21

2.1205 show pim ipv6 sparse-mode route count [...]

see 2.22

2.1206 show pim ipv6 sparse-mode rp [...]

see 2.23

2.1207 show pim ipv6 sparse-mode rp hash GROUP [...]

see 2.24

2.1208 show pim protocol counters [...]

see 2.681

2.1209 show pim sparse-mode evpn cli membership [...]

`vlans`	*table(numeric string) = VlanMembership* see 2.1209.1	Map VLAN ID to per-VLAN group membership information.

2.1209.1 VlanMembership

`[groups]`	*table(IPv4 or IPv6 address) = GroupMembership* see 2.1209.1.1	Map group to source membership information.
`[vrfName]`	*string*	VRF name.

2.1209.1.1 GroupMembership

`[sources]`	*table(IPv4 or IPv6 address) = SourceMembership* see 2.1209.1.1.1	Map Source to membership information.

2.1209.1.1.1 SourceMembership

`[mode]`	*enum: none | exclude | include*	Membership mode.

2.1210 show pim sparse-mode evpn gateway [...]

`vrfs`	*table(string) = EvpnGatewayVrf* see 2.1210.1	Mapping of VRFs to their VLAN IDs.

2.1210.1 EvpnGatewayVrf

`vlan`	*integer*	Internal VLAN ID for this VRF.

2.1211 show pim sparse-mode evpn pim membership [...]

see 2.1209

2.1212 show pim sparse-mode pim evpn membership [...]

vlans	*table(numeric string) = VlanMembership* see 2.1209.1	Map VLAN ID to per-VLAN group membership information.

2.1213 show platform (fap | arad) eci hardware-sync [...]

eciTimeSyncInfo	*table(string) = AradEciTimeSyncInfo* see 2.1213.1	Mapping of each Arad to its corresponding ECI TimeSync Information.

2.1213.1 AradEciTimeSyncInfo

latestCmicTod	*integer*	Latest CMIC ToD (in ns).
latestEciTod	*integer*	Latest ECI ToD (in ns).
numUpdates	*integer*	Number of ECI ToD updates received.
phase	*list of integer*	Last 4 phase differences of the ECI ToD (in ns).
state	*string*	State of Arad's ECI ToD.

2.1214 show platform (fap | arad) zuma [...]

zumaMappings	*table(string) = ZumaMapping* see 2.1214.1	A mapping between Arad names and their corresponding Zuma name and Microblaze index.

2.1214.1 ZumaMapping

microblazeIndex	*integer*	The microblaze index for this Arad.
zumaName	*string*	The name of the Zuma FPGA for this Arad.

2.1215 show platform (fap | arad) zuma detail [...]

zumasDetails	*table(string) = ZumaDetails* see 2.1215.1	A mapping between Zuma names and detailed information about those Zumas.

2.1215.1 ZumaDetails

aradsMicroblazeIndexes	*table(string) = integer*	A mapping between Arad names and their corresponding Microblaze index on this Zuma.
multiBitEccErrors	*integer*	The number of multi-bit ECC errors in this Zuma.
pciAddress	*string*	The PCI address of this Zuma.
singleBitEccErrors	*integer*	The number of single-bit ECC errors in this Zuma.

2.1216 show platform (fap | arad | jericho | jericho2) [...]

switches	*table(string) = FapInfo* see 2.1216.1	Maps FAP names to their corresponding chip information.

2.1216.1 FapInfo

[device]	*enum: n/A | qumran2a | jericho2m | arad | jericho3 | jerichoPlus | qumranAx | jericho2c | aradPlus | jericho2 | jericho2cPlus | jericho*	Chip device name.
[fapId]	*integer*	FAP ID.
[fapType]	*enum: n/A | arad | jericho3 | jericho2 | jericho*	Chip family name.
[revision]	*enum: n/A | 82a1 | 81a0 | a1 | q2c2tA1 | a2 | 84a0 | q2cPlusA1 | 81a1 | b0 | 82a0 | q2cPlusA2 | q2cPlusA0 | a0 | 84a1 | b1*	Chip revision name.

2.1217 show platform (fap | arad | jericho | jericho2) cint [...]

stages	*table(string) = CintFap* see 2.1217.1	A mapping between a stage and its faps.

2.1217.1 CintFap

faps	*table(string) = string*	A mapping between a fap and its cint.

2.1218 show platform (fap | arad | jericho | jericho2) copp mapping [...]

revision 2

classes	*table(string) = AradGenericCoppClassStats* see 2.1218.1	Generic COPP port information.
policerClasses	*table(string) = AradPolicerCoppClassStats* see 2.1218.2	Special COPP port information.

2.1218.1 AradGenericCoppClassStats

firstVoq	*integer*	First VOQ number.
minimumBandwidth	*integer*	COPP class minimum bandwidth in kbps.
shape	*integer*	COPP class shape in kbps.
sysPhyPort	*integer*	System Physical Port ID.
totalVoqs	*integer*	Total number of VOQs.

2.1218.2 AradPolicerCoppClassStats

`firstVoq`	*integer*	First VOQ ID.
`policerBurst`	*integer*	COPP class policer burst in kbytes.
`policerRate`	*integer*	COPP class policer rate in kbps.
`sysPhyPort`	*integer*	System physical port ID.
`totalVoqs`	*integer*	Total number of VOQs.

2.1219 show platform (fap | arad | jericho | jericho2) eedb outac [...]

faps	*table(string) = EedbOutAcPoolMap* see 2.1219.1	A mapping of FAP name to its output attachment circuit (OutAc) pools.

2.1219.1 EedbOutAcPoolMap

pools	*table(string) = EedbOutAcIndexMap* see 2.1219.1.1	A mapping of pool name to its OutAc entries.

2.1219.1.1 EedbOutAcIndexMap

[outAcIndices]	*table(numeric string) = EedbOutAcEntry* see 2.1219.1.1.1	A mapping of OutAc index to its OutAc details.

2.1219.1.1.1 EedbOutAcEntry

[bank]	*integer*	Bank used for the OutAc.
[bankOffset]	*string*	Bank and offset of OutAc.
[dropAction]	*boolean*	Drop packets when true.
[oamLif]	*boolean*	OAM LIF set.
[offset]	*integer*	Offset in the bank for the OutAc.
[outAcHexStr]	*string*	Output attachment circuit index.
[outAcIndex]	*integer*	Output attachment circuit index.
[outlifProfileId]	*integer*	OutLIF profile ID.
[pcpDeiProfileId]	*integer*	PCP-DEI profile ID.
[statObjectCmd]	*integer*	Statistics object command.
[statObjectId]	*integer*	Statistics object ID.
[sviUsers]	*integer*	SVI users.
[sviVlan]	*integer*	SVI VLAN ID.
[vid1]	*integer*	Outer VLAN tag.
[vid2]	*integer*	Inner VLAN tag.
[vlanEditProfileId]	*integer*	VLAN edit profile ID.

2.1220 show platform (fap | arad | jericho | jericho2) evpn mpls decap [...]

revision 2

[l2]	*L2EvpnMplsDecapMapping* see 2.1220.1	L2 decaps.
[l3]	*L3EvpnMplsDecapMapping* see 2.1220.2	L3 decaps.

2.1220.1 L2EvpnMplsDecapMapping

labels	*table(numeric string) = L2LabelBosEvpnMpls-DecapMapping* see 2.1220.1.1	A mapping of MPLS label to BOS flag states.

2.1220.1.1 L2LabelBosEvpnMplsDecapMapping

[bos0]	*L2EvpnMplsDecapEntries* see 2.1220.1.1.1	L2 hardware programming details when label BOS == 0.
[bos1]	*L2EvpnMplsDecapEntries* see 2.1220.1.1.1	L2 hardware programming details when label BOS == 1.

2.1220.1.1.1 L2EvpnMplsDecapEntries

decapEntries	*list of L2EvpnMplsDecapEntry* see 2.1220.1.1.1.1	Hardware programming detail of decap entries.

2.1220.1.1.1.1 L2EvpnMplsDecapEntry

[destination]	*integer*	Encoded destination.
[innerTag]	*integer*	Inner VLAN tag.
labelType	*enum: evpn | imet*	Label type.
lifBank	*integer*	Logical interface bank index.
lifIndex	*integer*	Logical interface index.
vlan	*integer*	VLAN.

2.1220.2 L3EvpnMplsDecapMapping

labels	*table(numeric string) = L3LabelBosEvpnMpls-DecapMapping* see 2.1220.2.1	A mapping of MPLS label to BOS flag states.

2.1220.2.1 L3LabelBosEvpnMplsDecapMapping

[bos0]	*L3EvpnMplsDecapEntry* see 2.1220.2.1.1	L3 hardware programming details when label BOS == 0.
[bos1]	*L3EvpnMplsDecapEntry* see 2.1220.2.1.1	L3 hardware programming details when label BOS == 1.

2.1220.2.1.1 L3EvpnMplsDecapEntry

lifBank	*integer*	Logical interface bank index.
lifIndex	*integer*	Logical interface index.
payloadType	*enum: mpls | ip*	Payload type.
rifIndex	*integer*	Routing interface index.
vrfName	*string*	VRF name.

2.1221 show platform (fap | arad | jericho | jericho2) evpn mpls encap [...]

revision 2

faps	*table(string) = VlanEvpnMplsEncapMap* see 2.1221.1	A mapping of FAP name to its VLAN IDs.

2.1221.1 VlanEvpnMplsEncapMap

vlans	*table(numeric string) = IntfEvpnMplsEncapMap* see 2.1221.1.1	A mapping of VLAN ID to its interfaces.

2.1221.1.1 IntfEvpnMplsEncapMap

interfaces	*table(Interface name) = EvpnMplsEncapEntries* see 2.1221.1.1.1	A mapping of interface name to its hardware programming details.

2.1221.1.1.1 EvpnMplsEncapEntries

intfEntries	*list of EvpnMplsEncapEntry* see 2.1221.1.1.1.1	Hardware programming detail of encap entry.

2.1221.1.1.1.1 EvpnMplsEncapEntry

[esiLabel]	*integer*	ESI label.
[lifBank]	*integer*	Logical interface bank index.
[lifIndex]	*integer*	Logical interface index.
[numConfiguredMembers]	*integer*	Count of configured members of the port-channel on the FAP.
[numProgrammedMembers]	*integer*	Count of programmed members of the port-channel on the FAP.
[tunnelId]	*integer*	Hardware tunnel ID.
[vlanDomain]	*integer*	VLAN domain.
[vlanTag]	*integer*	VLAN tag.

2.1222 show platform (fap | arad | jericho | jericho2) fabric load distribution [...]

faps	*table(string) = WfdConfigNumSerdes* see 2.1222.1	Load balancing config for the specified FAPs.

2.1222.1 WfdConfigNumSerdes

feNumSerdes	*table(string) = integer*	Show the number of links to each FE from this FAP.
[wfdProfile]	*integer*	Load balancing profile for this FAP.

2.1223 show platform (fap | arad | jericho | jericho2) fabric load distribution destination (fap | arad | jericho | jericho2) [...]

faps	*table(string) = FapWfdSerdesWeights* see 2.1223.1	Load balancing weights for these source FAPs to destination FAPs.

2.1223.1 FapWfdSerdesWeights

faps	*table(string) = WfdConfigSerdesWeights* see 2.1223.1.1	SerDes weights for destination FAPs given a source FAP.

2.1223.1.1 WfdConfigSerdesWeights

serdesWeights	*table(numeric string) = integer*	Weight from 1-127 for any given SerDes ID.
[wfdProfile]	*integer*	Load balancing profile for this FAP.

2.1224 show platform (fap | arad | jericho | jericho2) flow tracking sampled flow-table hw [...]

faps	*table(string) = FlowTableFapLevelInfo* see 2.1224.1	Flows per FAP.

2.1224.1 FlowTableFapLevelInfo

flows	*list of FlowTableEntry* see 2.1224.1.1	List of flows.

2.1224.1.1 FlowTableEntry

counterIndex	*integer*	Counter index.
dstAddr	*IP address*	Destination IP address.
dstPort	*integer*	Destination port.

fragmented	*boolean*	Whether to match on fragmented packets.
ipProtocol	*integer*	IP protocol.
portLabel	*integer*	Port label.
srcAddr	*IP address*	Source IP address.
srcPort	*integer*	Source port.
vlanId	*integer*	Flow VLAN ID.
vrf	*integer*	VRF.

2.1225 show platform (fap | arad | jericho | jericho2) ip route prefix compression tcam [...]

banks	*table(numeric string) = PrefixCompressionTcamEntries* see 2.1225.1	A mapping of TCAM bank to its indices.

2.1225.1 PrefixCompressionTcamEntries

indices	*table(numeric string) = PrefixCompressionTcamEntry* see 2.1225.1.1	A mapping of TCAM index to its entries.

2.1225.1.1 PrefixCompressionTcamEntry

destIpBit	*integer*	Destination IP bit indicator.
eeiMsb4b	*integer*	MSB 4 bits of EEI.
fec0	*integer*	FEC0 hardware index.
fec1	*integer*	FEC1 hardware index.
lem2Lookup	*integer*	LEM2 lookup success.
overridingFecId	*integer*	Overriding FEC hardware index.

2.1226 show platform (fap | arad | jericho | jericho2) label sharing inlif shadow [...]

revision 2

entries	*list of LabelSharingInLif-EntriesForFap* see 2.1226.1	Label sharing InLIF entries for each FAP.

2.1226.1 LabelSharingInLifEntriesForFap

entriesForFap	*list of LabelSharingInLifEntry* see 2.1226.1.1	Label sharing InLIF entries on this FAP.
[entriesForFapCore] **	*list of PmfDataInLifEntry* see 2.1226.1.2	PMF data InLif entries on this FAP and core. **Since: 4.23.2
fapId	*integer*	FAP ID.
[fapName] **	*string*	FAP name. **Since: 4.23.2
[localCoreId] **	*integer*	Local core ID. **Since: 4.23.2
[systemTcamId] **	*integer*	System TCAM ID. **Since: 4.23.2

2.1226.1.1 LabelSharingInLifEntry

[innerTag]	*integer*	Inner VLAN tag.
[lifEntry]	*LifAcMp* see 2.1226.1.1.3	LIF entry.
[outerTag]	*integer*	Outer VLAN tag.
request	*LabelSharingRequest* see 2.1226.1.1.2	Label sharing request.
[semPayload]	*InAcSemPayload* see 2.1226.1.1.1	SEM payload.
[vlanDomain]	*integer*	VLAN domain.

2.1226.1.1.1 InAcSemPayload

lifIndex	*integer*	Raw SEM payload value converted to SW LIF index.
startingLifBank	*integer*	Starting LIF bank index.
value	*integer*	Raw U32 sem payload value.

2.1226.1.1.2 LabelSharingRequest

fapId	*integer*	The FAP ID of the FAP this entry was found on.
intfName	*interface name*	Interface name.
l3FloodPortVlanId	*integer*	Layer 3 flood port VLAN ID.

2.1226.1.1.3 LifAcMp

inLifAdditionalData	*InLifAdditionalData* see 2.1226.1.1.3.1	InLifAdditionalData value.
vsi	*integer*	VSI value.

2.1226.1.1.3.1 InLifAdditionalData

ipv4	*integer*	IPv4 InLifAdditionalData field.
ipv6	*integer*	IPv6 InLifAdditionalData field.
l2PF **	*integer*	l2PF InLifAdditionalData field. **Since: 4.26.0
macl	*integer*	MACL InLifAdditionalData field.
pbr	*integer*	PBR InLifAdditionalData field.
policerIdxMeterA **	*integer*	policerIndexMeterA InLifAdditionalData field. **Since: 4.26.0
policerIdxMeterB **	*integer*	policerIndexMeterB InLifAdditionalData field. **Since: 4.26.0
qos	*integer*	QoS InLifAdditionalData field.
sFlow **	*integer*	sflowSubintf InLifAdditionalData field. **Since: 4.29.0
value	*integer*	Raw InLifAdditionalData U64 value.

2.1226.1.2 PmfDataInLifEntry

[innerTag]	*integer*	Inner VLAN tag.
intfName	*interface name*	Interface name.
[lagName]	*interface name*	Parent LAG interface name.
[lifEntry]	*LifAcMpGenericData* see 2.1226.1.2.1	LIF entry.
[lifIndex]	*integer*	Raw SEM payload value converted to SW LIF index.
[outerTag]	*integer*	Outer VLAN tag.
[vlanDomain]	*integer*	VLAN domain.
[vlanIntfName] **	*interface name*	VLAN interface name. **Since: 4.24.1

2.1226.1.2.1 LifAcMpGenericData

eei	*integer*	Egress Encapsulation Information.
lifGenericData0	*integer*	LIF generic data word 0.
lifGenericData1	*integer*	LIF generic data word 1.
policyId	*PolicyId* see 2.1226.1.2.1.1	Policy ID for every feature type.

2.1226.1.2.1.1 PolicyId

[lifAclBlockIntf] **	*integer*	BlockIntf label **Since: 4.26.2
[lifAegisLabel]	*integer*	AEGIS label
[lifAegisLabelVlan-Interface] **	*integer*	AEGIS VLAN Interface label **Since: **currentRelease**
[lifAegisUnsharedPolicer] **	*integer*	Base label for unshared Aegis policers **Since: 4.29.1
[lifArpFeature] **	*integer*	ARP feature label **Since: 4.26.2
[lifBumFilter] **	*integer*	EVPN MPLS BUM filter label **Since: 4.27.2
[lifCpuLabel] **	*integer*	CPU policy label **Since: 4.24.1
[lifDhcpSnooping] **	*integer*	DHCP snooping label **Since: 4.29.1
[lifEsiLabel]	*integer*	ESI label
[lifEsiLabelTunnelId] **	*integer*	ESI label and HwTunnel Id **Since: 4.30.1
[lifFlowspecLabel] **	*integer*	Flowspec label **Since: 4.24.1
[lifGroupPolicing] **	*integer*	Policer label for grouped interface policing **Since: 4.29.1
[lifInterfacePolicing] **	*integer*	Dedicated policer label for interface policing **Since: 4.29.1
[lifIpsecLabel] **	*integer*	IPsec label **Since: 4.29.1
[lifJ2GroupPolicing] **	*integer*	Policer label for grouped interface policing **Since: 4.30.0
[lifJ2InterfacePolicing] **	*integer*	Dedicated policer label for interface policing **Since: 4.28.2
[lifL2ProtocolFwdLabel]	*integer*	L2 protocol forwarding label
[lifL2SubIntf] **	*integer*	L2 Subinterface label **Since: 4.27.2
[lifMaclLabel]	*integer*	MACL label
[lifMplsEvpnLabel1]	*integer*	MPLS EVPN label 1
[lifMplsEvpnLabel2]	*integer*	MPLS EVPN label 2
[lifPaclV4Label] **	*integer*	IPv4 PACL label **Since: 4.24.0
[lifPaclV6Label] **	*integer*	IPv6 PACL label **Since: 4.24.0
[lifPbrLabel]	*integer*	PBR label
[lifPvlan] **	*integer*	PVLAN secondary VLAN **Since: 4.29.2
[lifPwxOam] **	*integer*	Label for L3 subinterfaces patched to pseudowire **Since: 4.29.1
[lifQosLabel]	*integer*	QoS label
[lifRaclV4Label] **	*integer*	IPv4 RACL label **Since: 4.28.0
[lifRaclV6Label]	*integer*	IPv6 RACL label

`[lifSampledFlowTracking-Label]` **	*integer*	Sampled flow tracking label **Since: 4.25.0
`[lifSegmentSecurity]` **	*integer*	Segment Security label **Since: 4.29.1
`[lifSflowSubintf]` **	*integer*	Sflow label for subinterfaces **Since: 4.29.1
`[lifStormControlSubintf]` **	*integer*	StormControl label for subinterfaces **Since: **currentRelease**
`[lifTunnelAclV4]` **	*integer*	IPv4 ACL label for GRE/IPsec tunnel interfaces **Since: 4.29.1
`[lifTunnelPbrV4]` **	*integer*	IPV4 PBR label for GRE/IPsec tunnel interfaces **Since: 4.29.1
`[lifUnsharedPolicer]` **	*integer*	Base policer label for unshared QoS policy-maps **Since: 4.28.2
`[lifUrpfLookupVrfLabel]` **	*integer*	uRPF lookup VRF label **Since: 4.28.2
`[lifVlanEdit]` **	*integer*	Label for extended vlan editing **Since: 4.29.1
`[lifVlanEtreeRole]` **	*integer*	VLAN etree role label **Since: 4.27.2
`[lifVrfSelectionLabel]` **	*integer*	VRF Selection Policy label **Since: 4.28.1

2.1227 show platform (fap | arad | jericho | jericho2) label sharing isem (shadow | hardware | diff) [...]

isemForFap	*list of LabelSharingIsemForFap* see 2.1227.1	Label sharing ISEM for each FAP.

2.1227.1 LabelSharingIsemForFap

entriesWithRequests	*list of LabelSharingIsemEntryWithRequest* see 2.1227.1.2	Label sharing requests and ISEM entries inserted.
entriesWithRequests-Mismatches	*list of LabelSharingIsemEntryWithRequestMismatch* see 2.1227.1.3	Label sharing requests and ISEM entries inserted that mismatch.
entryOnly	*list of LabelSharingIsemEntryOnly* see 2.1227.1.1	Label sharing ISEM entry with no matching request.
fapId	*integer*	FAP ID.
isHalo	*boolean*	True if this FAP is Halo enabled.
requestOnly	*list of LabelSharingRequestOnly* see 2.1227.1.4	List of label sharing requests that are in progress.

2.1227.1.1 LabelSharingIsemEntryOnly

policyId	*LabelSharingPolicyId* see 2.1227.1.1.1	Policy ID parsed from ISEM.
vlanDomain	*integer*	VLAN domain.
vlanId	*integer*	VLAN ID.

2.1227.1.1.1 LabelSharingPolicyId

haloAclLabel	*integer*	Halo ACL label bits.
ipv4Policy	*integer*	IPv4 policy ID bits.
ipv6EgressPolicy	*integer*	IPv6 egress policy bits.
ipv6Policy	*integer*	IPv6 policy ID bit.
ipv6PolicyValid	*integer*	IPv6 policy valid bit.
pbrPolicy	*integer*	PBR policy ID bits.
qosPolicy	*integer*	QoS policy ID bits.
value	*integer*	Raw policy ID value.

2.1227.1.2 LabelSharingIsemEntryWithRequest

policyId	*LabelSharingPolicyId* see 2.1227.1.1.1	Policy ID parsed from ISEM.
request	*LabelSharingRequest* see 2.1226.1.1.2	Label sharing request.
vlanDomain	*integer*	VLAN domain.
vlanId	*integer*	VLAN ID.

2.1227.1.3 LabelSharingIsemEntryWithRequestMismatch

expected	*LabelSharingPolicyId* see 2.1227.1.1.1	The requested policy ID.
policyId	*LabelSharingPolicyId* see 2.1227.1.1.1	Policy ID parsed from ISEM.
request	*LabelSharingRequest* see 2.1226.1.1.2	Label sharing request.
vlanDomain	*integer*	VLAN domain.
vlanId	*integer*	VLAN ID.

2.1227.1.4 LabelSharingRequestOnly

expected	*LabelSharingPolicyId* see 2.1227.1.1.1	Policy ID requested to be inserted.
request	*LabelSharingRequest* see 2.1226.1.1.2	Label sharing request.
[vlanDomain]	*integer*	VLAN domain.
[vlanId]	*integer*	VLAN ID.

2.1228 show platform (fap | arad | jericho | jericho2) label sharing ppmc l2 shadow [...]

ppmcEntries	*table(numeric string) = PpmcLabelSharingEntries* see 2.1228.1	A mapping of PPMC index to PPMC label sharing entry.

2.1228.1 PpmcLabelSharingEntries

globalOutlif1	*integer*	Global Outlif1.
globalOutlif2	*integer*	Global Outlif2.
[ppmcMacLabel]	*integer*	Egress Mac label.
[ppmcV4Label]	*integer*	Egress IPv4 label.
[ppmcV6Label]	*integer*	Egress IPv6 label.

2.1229 show platform (fap | arad | jericho | jericho2) label sharing ppmc l3 shadow [...]

see 2.1228

2.1230 show platform (fap | arad | jericho | jericho2) lag [...]

revision 2

aradLags	*table(string) = AradLags* see 2.1230.1	Mapping of Arad chip name to LAG information.

2.1230.1 AradLags

lags	*table(string) = AradLag* see 2.1230.1.1	Mapping of port channel name to LAG ID and associated port lists.
mode	*enum: 512x32 | 2048x8 | 64x256 | 256x64 | 1024x16 | 128x128*	Programmed LAG mode.

2.1230.1.1 AradLag

collectingMemberOrder	*list of Interface name*	Order of interface.
collectingMembers	*table(Interface name) = integer*	Mapping of interface name to sysPortId.
distributingMemberOrder	*list of Interface name*	Order of interface.
distributingMemberWeights **	*table(Interface name) = integer*	Mapping of interface name to speed weight. **Since: 4.26.0
distributingMembers	*table(Interface name) = integer*	Mapping of interface name to sysPortId.
id	*string*	Hardware LAG ID.
unmappedCollectingMembers **	*list of integer*	SysPortId of members who don't have an interface name. **Since: 4.17.0
unmappedDistributingMembers **	*list of integer*	SysPortId of members who don't have an interface name. **Since: 4.17.0

2.1231 show platform (fap | arad | jericho | jericho2) mapping [...]

revision 5

aradMappings	*list of PortMappings* see 2.1231.1	List of Port Mappings for each switch.

2.1231.1 PortMappings

baseSystemCoreId	*integer*	System Core Id of the first core of this fap.
cores	*integer*	Number of cores for this fap.
fapId	*integer*	Fap Id.
fapName	*string*	Fap Name.
portMappings	*table(numeric string) = PortMapping* see 2.1231.1.1	Maps Port Mappings to the system physical port IDs.

2.1231.1.1 PortMapping

baseQueuePair	*integer*	First egress transmit queue pair used by the system port.
coreId	*integer*	Core Id.
fapId	*integer*	Fap Id.
fapPortId	*integer*	Fap Port Id.
interface	*interface name*	An Interface name that maps to the system port.
nifPortId	*integer*	Nif Port Id.
outTmPortId	*integer*	Out TM Port Id.
qsgmiiId	*integer*	QSGMII ID used by the system port.
queuePairs	*integer*	Number of queue pairs used by the port in the egress.
serdesIds	*list of integer*	List of Serdes used by the system port.
voqId	*integer*	Voq Id.
xlgeId	*integer*	40G xlge which maps to the system port.

2.1232 show platform (fap | arad | jericho | jericho2) mroute [...]

[cmdErr]	*string*	Error Message.
[ipv4]	*FapToIpv4MroutesMapping* see 2.1232.1	Model for IPv4 Multicast Mroutes within respective FAP.
[ipv6]	*FapToIpv6MroutesMapping* see 2.1232.2	Model for IPv6 Multicast Mroutes within respective FAP.

2.1232.1 FapToIpv4MroutesMapping

[cmdErr]	*string*	Error Message.
[faps]	*table(string) = Ipv4MroutesList* see 2.1232.1.1	Mapping FAP name to its IPv4 multicast routes entries.

2.1232.1.1 Ipv4MroutesList

multicastRoutes	*list of Ipv4MrouteModel* see 2.1232.1.1.1	List of detailed IPv4 multicast routes entries.

2.1232.1.1.1 Ipv4MrouteModel

[cmdErr]	*string*	Error Message.
[group]	*IP address prefix*	Group ID.
[iifVlan]	*integer*	Incoming interface VLAN.
[iifVrf]	*integer*	Incoming interface VRF.
[inLem]	*boolean*	Mroute installed in LEM table.
[instance]	*integer*	TCAM Bank instance of multicast route entry.
[location]	*integer*	Index of mroute entry in TCAM.
[multicastId]	*integer*	Multicast ID.
[oifCpu]	*boolean*	OIF have Ingress Receive Editor, in packet goes to CPU too.
[oifVlans]	*table(numeric string) = OutgoingInfo* see 2.1232.1.1.1.1	Mapping OIF VLAN to its Interfaces.
[source]	*IP address prefix*	Source IP Prefix.

2.1232.1.1.1.1 OutgoingInfo

interfaces	*list of Interface name*	Outgoing interfaces.
[isPmsi] **	*boolean*	MVPN encap outgoing interface. **Since: 4.27.1
[tunnelEndInterfaces]	*list of Interface name*	Core facing interfaces: multicast packet copies for these Interfaces are done with encapsulation.

2.1232.2 FapToIpv6MroutesMapping

[cmdErr]	*string*	Error Message.
[faps]	*table(string) = Ipv6MroutesList* see 2.1232.2.1	Mapping FAP name to its IPv6 multicast routes entries.

2.1232.2.1 Ipv6MroutesList

multicastRoutes	*list of Ipv6MrouteModel* see 2.1232.2.1.1	List of detailed IPv6 multicast routes entries.

2.1232.2.1.1 Ipv6MrouteModel

[cmdErr]	*string*	Error Message.
[flpGroupId]	*integer*	FLP Group ID.
[flpLocation]	*integer*	FLP Location.
[group]	*IP address prefix*	Group ID in TT stage.
[iifVlan]	*integer*	Incoming interface VLAN.
[instance] **	*integer*	TCAM Bank instance of multicast route entry. **Since: 4.27.1
[location] **	*integer*	Index of mroute entry in TCAM. **Since: 4.27.1
[multicastId]	*integer*	Multicast ID.
[oifCpu]	*boolean*	OIF have Ingress Receive Editor, in packet goes to CPU too.
[oifVlans]	*table(numeric string) = OutgoingInfo* see 2.1232.1.1.1.1	Mapping OIF VLAN to its Interfaces.
[source]	*IP address prefix*	Source IP Prefix.
[ttGroupId]	*integer*	TT Group ID.
[ttLocation]	*integer*	TT Location.

2.1233 show platform (fap | arad | jericho | jericho2) mroute ipv4 [...]

see 2.1232.1

2.1234 show platform (fap | arad | jericho | jericho2) mroute ipv6 [...]

see 2.1232.2

2.1235 show platform (fap | arad | jericho | jericho2) mvpn decap [...]

labels	*table(numeric string) = MvpnDecapEntry* see 2.1235.1	A mapping of MPLS label to MVPN decap information.

2.1235.1 MvpnDecapEntry

lifIndex	*integer*	Logical interface index.
pmsiVlan	*integer*	PMSI VLAN interface ID.

2.1236 show platform (fap | arad | jericho | jericho2) phy PHY_FROM_PORT dump [...]

aradPhysicalInfo	*list of PhysicalInfo* see 2.1236.1	The Arad Physical information.

2.1236.1 PhysicalInfo

aradName	*string*	Arad Name.
ports	*list of string*	Physical information for the ports.

2.1237 show platform (fap | arad | jericho | jericho2) phy PHY_FROM_PORT register mmd PHY_REG_START [...]

aradPhysicalRegisters	*list of PhysicalRegister* see 2.1237.1	The Arad Physical registers.

2.1237.1 PhysicalRegister

aradName	*string*	Arad Name.
ports	*list of string*	Physical register information for the ports.

2.1238 show platform (fap | arad | jericho | jericho2) pmf [...]

profileInstances	*list of ProfileInstance* see 2.1238.1	List of profile instances.

2.1238.1 ProfileInstance

fapIds	*list of integer*	Programmed FAP IDs.
fapRevMajor	*enum: majorJericho2 | majorJericho3 | majorJericho2cPlus | majorJerichoPlus | majorJericho2c | majorArad | majorUnknown | majorQumran2a | majorJericho2m | majorJericho | majorQumranAx | majorAradPlus*	Major FAP revision.
fapRevMinor	*enum: minorQ2cPlusA0 | minor84A0 | minorA1 | minorUnknown | minorA0 | minorA2 | minorQ2c2tA1 | minorB0 | minor84A1 | minorB1 | minor81A0 | minor82A1 | minor82A0 | minorQ2cPlusA1 | minorQ2cPlusA2 | minor81A1*	Minor FAP revision.
multiAgent	*boolean*	SandFap/SandPmf both programming PMF.
profileName	*string*	TCAM profile name.
stages	*table(string) = Stage* see 2.1238.1.1	Mapping from PMF stage to preselectors and contexts.
version	*integer*	TCAM profile version number.

2.1238.1.1 Stage

contexts	*table(numeric string) = Context* see 2.1238.1.1.1	Mapping from context ID to key and action info.
preselectors	*table(string) = Preselector* see 2.1238.1.1.2	Mapping from preselector ID to match criteria.

2.1238.1.1.1 Context

fesIds	*table(numeric string) = Fes* see 2.1238.1.1.1.1	Mapping from FES ID to action info.
keyConstructions	*table(string) = KeyConstruction* see 2.1238.1.1.1.2	Mapping from PmfKey to key construction info.
keyLookups	*table(string) = KeyLookup* see 2.1238.1.1.1.4	Mapping from PmfKey to key lookup info.

2.1238.1.1.1.1 Fes

andMask	*integer*	AND-mask applied to FES source.

db	*enum: dbSystem6* | *dbCbf* | *dbDirectExtractionSet-UnderlaySysInLif* | *dbTrafficPolicyIp6* | *dbDirectExtractionPmf1-GroupPolicer* | *dbDummy* | *dbEgressIpAcl* | *dbTrafficPolicyIp4Egress* | *dbIpCpuTrafficPolicy* | *dbL4SrcPortXformed* | *dbIpRacl* | *dbIp6SubIntfAcl* | *dbMplsSystem* | *dbIpTapAgg* | *dbElkResultIpPacl* | *dbIpMcastRouting* | *swEgressIpBridge* | *dbIpSrcXformed* | *dbIpUcastRouting* | *dbEgressMacAcl* | *dbTcCounters* | *dbNone* | *dbIpSubIntfAcl* | *dbVxlanRouting* | *dbElkIp6* | *swInnerIpOGre* | *dbVxlan0* | *dbIpsec* | *dbIpv6MpTt2ndPass* | *dbIpQos* | *dbDirectExtractionPmf1-IntfPolicingCounter* | *dbDirectExtractionPmf3* | *dbMacMirror* | *dbInAc* | *db-VxlanFemDirectExtraction* | *dbDirectExtractionPmf1-StormControlSubIntf* | *dbMplsSubIntfPbr* | *dbEvpnMplsBridging* | *dbIngressEgressCascade* | *dbEgressVxlanDrop* | *dbVniLookup* | *dbElkResultIp6Racl* | *dbEvpnMplsEsiFiltering* | *dbSystemMac* | *dbIp6RaclStage2* | *dbEvpn-MplsEsiLocalToRemote* | *dbTunnelTermination* | *dbIpTunnelIntfPbr* | *dbIp6SubIntfQos* | *dbIpv6VrfDb* | *dbMacTapAgg* | *dbDirect-ExtractionPmf3Srv6* | *dbIp6Mirror* | *dbEgressIp6Racl* | *dbFemDirectExtraction* | *dbIp6Pbr* | *dbRecircEgressTrap* | *dbIp6Qos* | *dbVxlanMlag* | *dbIp6Pacl* | *dbDirectExtractionSetEcn* | *dbDirectExtractionPmf1-Srv6* | *dbMacQos* | *swIpOMpls* | *dbIpv4TtBasic* | *dbTapPort* | *dbMplsFrr* |	Corresponding Database for source bits.

hwAction	*enum: oamIdAct* | *generalData31To00Act* | *dpMeterCommandAct* | *mirrorAct* | *wexemLearnKey2Act* | *ftmhDpAct* | *learnInfoKey1Act* | *tracePacketAct* | *bierStringSizeAct* | *ebtaEopEpmfDeltaAct* | *statisticsObject0Act* | *mirrorProfileAct* | *statisticsObject11Act* | *udh127To96Act* | *statisticsObject9Act* | *dspaAct* | *bierStringOffsetAct* | *learnOrTransplantAct* | *forwardAct* | *tunnelPriorityAct* | *slbFoundAct* | *statisticsObject10Act* | *dualQueueAct* | *extStatistics3Act* | *statisticsObjectAtr0Act* | *userHeader2TypeAct* | *tcAct* | *slbKey1Act* | *vlanEditCommandAct* | *rpfDestinationAct* | *pmfSnoopStrengthAct* | *userHeader4Act* | *learnInfo2Act* | *destinationAct* | *vlanEditPcpDei1Act* | *outLif3Act* | *pmfCounter1PtrAct* | *outLifPushAct* | *counter1PtrAct* | *outTmPortAct* | *learnInfo1Act* | *wexemLearnPayload1Act* | *ftmhAse31To00Act* | *ingressTimeStampOverrideAct* | *bytesToRemoveAct* | *wexemLearnNeededAct* | *ftmhAqmProfileAct* | *aceStatMeterAct* | *statisticsObject6Act* | *statisticsObjectAtr6Act* | *learnInfoFormatIndexAct* | *wexemLearnAppdbAct* | *outLifNextIndexAct* | *pmfEesEntry119To90Act* | *pmfCounter0ProfileAct* | *pmfEesEntry59To30Act* | *aclContextAct* | *forwardActionDestinationAct* | *cosProfileAct* | *pemGeneralData0Act* | *counter0PtrAct* | *lagLbKeyActMetadata2* |	Hardware action.

`instruction`	*integer*	FES 2nd instruction ID.
`orMask`	*integer*	OR-mask applied to FES source.
`orSelect`	*integer*	Source for OR-mask.
`polarity`	*integer*	If validBit is true, polarity of LSB.
`shift`	*integer*	Right bit-shift applied to FES source.

source	*enum:*	Source bits for FES. *pmf2_lexem_27_0_4_b0_tcam_7_63_32* | *pmf3_direct_87_24* | *pmf2_tcam_5_31_0_tcam_4_63_32* | *pmf2_direct_1_87_24* | *pmf2_direct_1_119_56* | *pmf2_tcam_6_31_0_tcam_5_63_32* | *pmf2_tcam_1_31_0_tcam_0_63_32* | *pmf2_key_j_159_96* | *pmf3_sexem_59_0_4_b0* | *pmf2_direct_0_23_0_8_b0_key_j_159_128* | *pmf3_key_a_127_64* | *pmf3_tcam_1_63_0* | *pmf2_tcam_7_63_0* | *pmf2_tcam_6_63_0* | *pmf2_tcam_4_63_0* | *pmf2_key_j_63_0* | *pmf2_direct_1_119_88* | *pmf2_key_j_95_32* | *pmf2_tcam_2_31_0_tcam_1_63_32* | *pmf2_key_i_127_64* | *pmf2_direct_0_87_24* | *pmf2_tcam_5_63_0* | *pmf2_direct_0_55_0_8_b0* | *egress_tcam_1_payload_31_0_tcam_0_payload_63_32* | *pmf3_key_b_127_64* | *pmf2_key_j_127_64* | *pmf2_direct_1_55_0_8_b0* | *pmf2_tcam_7_31_0_tcam_6_63_32* | *egress_tcam_0_payload_63_0* | *pmf2_sexem_59_0_4_b0* | *pmf3_direct_119_56* | *pmf2_tcam_2_63_0* | *pmf3_tcam_0_63_0* | *pmf2_tcam_3_31_0_tcam_2_63_32* | *pmf2_sexem_27_0_4_b0_lexem_59_28* | *pmf2_key_i_95_32* | *pmf3_direct_55_0_8_b0* | *pmf2_tcam_1_63_0* | *pmf2_key_i_63_0* | *pmf2_direct_1_23_0_8_b0_direct_0_119_88* | *pmf2_tcam_4_31_0_tcam_3_63_32* | *pmf3_key_b_159_96* | *pmf3_key_a_31_0_sexem_59_28* | *pmf2_tcam_0_63_0* | *pmf2_tcam_3_63_0* | *pmf3_key_a_159_96* | *pmf2_key_i_159_96* | *pmf2_key_i_31_0_sexem_59_28* | *pmf3_key_b_63_0* | *pmf2_key_j_31_0_key_i_159_128* | *pmf3_direct_31_0_key_b_159_128* | *pmf2_direct_0_119_56* | *pmf3_key_a_95_32* | *egress_lexem_4_b0_59_0* | *egress_tcam_1_payload_63_0*

tcamAction	*enum: learn_sa_16_to_47* | *femTrafficPolicyUnshared-Policer* | *statId0* | *in_dscp_exp* | *egress_learn_enable* | *statProfile5* | *set_ecn1* | *vlan_edit_vid_2* | *userHeader3Type* | *permitLog* | *sys_in_lif* | *userHeader3* | *compressed_drop* | *statProfile4* | *default_compressed_drop* | *udh1AlwaysOn* | *vportNew* | *femDropDirectFlow* | *statId7* | *femPolicerData* | *femFwdHdrEncap* | *femSviCounter* | *dp* | *femTagRemoval* | *femVsi2* | *statId3* | *ace_type* | *ecmp_lb* | *parsingStartOffset* | *femDropTrafficPolicy* | *fwdLayerAdditionalInfo* | *egrStatProfile1* | *femVsi1* | *cudOutlifOrMcdbPtr* | *femCopyMeterBIndex* | *bytes_to_remove* | *femDropPacl* | *femDropIp6TapAgg* | *udh2-TypeAlwaysOnSlMinusOne* | *trap* | *dest* | *learn_sa_0_to_15* | *femVsi* | *inlif_profile_reduced* | *vxlan_decap_type* | *latencyFlow* | *statId2* | *userHeaderType* | *femCpuItmhOam* | *udh2AlwaysOn* | *femDrop* | *femMogTtlCopy* | *statId1* | *ieee_1588* | *egrCounterPtr* | *fwdDomain* | *femExpCopy* | *nwkQos* | *default_egrDiscard* | *femLem2Fec* | *native_vsi* | *statProfileIntfPolCounter* | *egr_trap* | *action_unknown* | *trap_strength* | *statId* | *action_eei* | *egrStatProfile* | *vlan_edit_command* | *userHeader4* | *femTtlCopy* | *pphType* | *statProfile6* | *egrDiscard* | *vlan_edit_vid_1* | *statId6* | *statProfile9* | *fwd_offset* | *outlif0_17b* | *userHeader2Type* | *mirror* | *femVsi3* | *femVtepDecapCounter* | *statProfile3* | *systemHeaderSizeAdjust* | *egrCounterPtr0* |	Action SW label.

validBit	*boolean*	Whether LSB of FES source enables action.

2.1238.1.1.1.2 KeyConstruction

db	*enum: dbSystem6* | *dbCbf* | *dbDirectExtractionSet-UnderlaySysInLif* | *dbTrafficPolicyIp6* | *dbDirectExtractionPmf1-GroupPolicer* | *dbDummy* | *dbEgressIpAcl* | *dbTrafficPolicyIp4Egress* | *dbIpCpuTrafficPolicy* | *dbL4SrcPortXformed* | *dbIpRacl* | *dbIp6SubIntfAcl* | *dbMplsSystem* | *dbIpTapAgg* | *dbElkResultIpPacl* | *dbIpMcastRouting* | *swEgressIpBridge* | *dbIpSrcXformed* | *dbIpUcastRouting* | *dbEgressMacAcl* | *dbTcCounters* | *dbNone* | *dbIpSubIntfAcl* | *dbVxlanRouting* | *dbElkIp6* | *swInnerIpOGre* | *dbVxlan0* | *dbIpsec* | *dbIpv6MpTt2ndPass* | *dbIpQos* | *dbDirectExtractionPmf1-IntfPolicingCounter* | *dbDirectExtractionPmf3* | *dbMacMirror* | *dbInAc* | *db-VxlanFemDirectExtraction* | *dbDirectExtractionPmf1-StormControlSubIntf* | *dbMplsSubIntfPbr* | *dbEvpnMplsBridging* | *dbIngressEgressCascade* | *dbEgressVxlanDrop* | *dbVniLookup* | *dbElkResultIp6Racl* | *dbEvpnMplsEsiFiltering* | *dbSystemMac* | *dbIp6RaclStage2* | *dbEvpn-MplsEsiLocalToRemote* | *dbTunnelTermination* | *dbIpTunnelIntfPbr* | *dbIp6SubIntfQos* | *dbIpv6VrfDb* | *dbMacTapAgg* | *dbDirect-ExtractionPmf3Srv6* | *dbIp6Mirror* | *dbEgressIp6Racl* | *dbFemDirectExtraction* | *dbIp6Pbr* | *dbRecircEgressTrap* | *dbIp6Qos* | *dbVxlanMlag* | *dbIp6Pacl* | *dbDirectExtractionSetEcn* | *dbDirectExtractionPmf1-Srv6* | *dbMacQos* | *swIpOMpls* | *dbIpv4TtBasic* | *dbTapPort* | *dbMplsFrr* |	PMF database.

`ffcIds`	*table(numeric string) = Ffc* see 2.1238.1.1.1.3	Mapping from FFC ID to FFC attributes.

2.1238.1.1.1.3 Ffc

hwQualifier	*enum: pmf1ExemAction* | *ipv63rdAddHeaderExists* | *aclContext* | *pmf1TcamAction2* | *ecmpLbKey1* | *pmf1TcamAction3* | *localOutLif* | *keyHInitial* | *statisticsAtr11* | *ipv42ndAddHeaderExists* | *ferStage2ResolvedFecPtr* | *btcMinusBtk* | *userHeader3Type* | *ethIsEth1* | *udhBaseEntryPushProfile* | *bierStrOffset* | *generalData* | *vlanEditPcpDei2* | *statisticsId11* | *rpfEcmpMode* | *cmpKey1Decoded* | *elephantLearnNeeded* | *vidValid* | *pphFheiSize* | *l4opsEncoder0* | *fwdLayerAdditionalInfo* | *statisticsId2* | *cudOutlifOrMcdbPtr* | *isTdm* | *ferStage0InDst* | *visibility* | *ftmhTmProfile* | *statisticsMetaData* | *extStatisticsAtr1* | *oamStampOffset* | *tmProfile* | *outLifProfile* | *ftmhAqmProfile* | *allOnes* | *pphTtl* | *fwdActionProfileIndex* | *pphValue1* | *mplsNofLabels* | *wexemKey0* | *ferStage1ResolvedFecPtr* | *ptc* | *pweInLifWideData* | *admtProfile* | *vlanEditPcp2* | *ftmhEcnEnable* | *ftmhDp* | *ferStage2InDst* | *prtQualifier* | *learnVsi* | *itmhPphType* | *ipv4FragmentedNonFirst* | *unknownAddress* | *ftmh-McidOrOutlif0OrMcdbPtr* | *statisticsId0* | *pphParsingStartType* | *globInLif0* | *pmf1TcamAction1* | *stateData* | *elkLkpPayloadAllPart1* | *oamId* | *eth2ndTpidIndex* | *discard* | *statisticsId8* | *wexemLearnAppdb* | *vlanEditVid2* | *ftmhBierBfrExtPresent* | *ftmhTmDestinationExt-Present* | *statisticsId9* | *ptcKeyGenVar* | *btkOffsetSop* |	Hardware qualifier.

index	*integer*	Header/layer index for base offset.
lsb	*integer*	Key offset LSB.
msb	*integer*	Key offset MSB.
offset	*integer*	Bit offset from base header/layer.
size	*integer*	Number of bits extracted by FFC.
source	*enum: absoluteLayerRecord | metaData | headerOrMetadata | absoluteHeader | forwardingHeader | layerRecord | cascade*	FFC source.

tcamQualifier	*enum:* *erpp_src_system_port* | *erpp_in_lif_orientation* | *tt_isem_a_hit* | *fwd_hdr_ipv6_hop_limit* | *inner_ipv6_sip_high_32* | *ipv6_dip_low_60_63* | *ipv6_tc* | *hdr_4_ipv6_dip_high_32* | *hdr_4_ipv4_dip_low* | *parsed_next_prtcl* | *vxlan_postcard_marker_vxlan* | *mpls_exp* | *key_changed_padded* | *fwd_hdr_ipv4_tos* | *tcp_first_option_kind* | *fwd_hdr_offset_value_32* | *in_lif_data_pbr* | *fwd_hdr_offset_value_0* | *dsp_learn_format* | *inner_vlan_tag_id* | *in_port_pmf3_aegis* | *ipv6_mc* | *esi_label* | *hdr_4_ipv4_sip_high* | *inner_ipv4_next_prtcl* | *fwd_hdr_ipv4_fragmented* | *in_port_qos* | *in_lif_cpu_policy* | *in_lif_data_intfMeter_idx* | *mplsogre_ip6_dip_low_32* | *erpp_ppmc_mac_acl_label* | *l4ops_6b* | *fwd_hdr_parsed_next_prtcl* | *in_port_qos_policer* | *erpp_ppmc_ipv4_acl_label* | *hdr_3_ipv4_dip_high* | *ll_mirror_profile* | *in_port_ip6_pbr* | *dp_and_tm_action_type* | *in_lif_data_basePolicer_idx* | *in_lif_data_l2_subintf_arp* | *hdr_4_ip_frag_offset* | *erpp_nwk_rcrd_ipv4_l4_src_port* | *ieee1588_encapsulation* | *vt_lookup1_payload* | *in_ttl* | *in_lif_segment_security* | *vt_lookup1_ip4* | *hdr_4_ipv4_tos* | *mplsogre_ip6_dip_high_0* | *in_lif_data_prefix* | *in_port_cpu_policy6* | *inner_ip_frag_offset* | *hdr_5_ipv4_src_port* | *egq_l4_src_dst_port* | *mpls_bos_payload_type* | *vxlan_inner_ip6_ttl_sub* | *is_ptp* | *inner_ipv4_sip* | *in_port_ip_pbr* | *hdr_4_ipv4_mc_opt_frag* | *mplsogre_mpls_ip_tos* | *ipv4_sip_low_8b* |	Qualifier SW label.

2.1238.1.1.1.4 KeyLookup

appDb	*integer*	AppDbId for PMF key.

db	*enum: dbSystem6* | *dbCbf* | *dbDirectExtractionSet-UnderlaySysInLif* | *dbTrafficPolicyIp6* | *dbDirectExtractionPmf1-GroupPolicer* | *dbDummy* | *dbEgressIpAcl* | *dbTrafficPolicyIp4Egress* | *dbIpCpuTrafficPolicy* | *dbL4SrcPortXformed* | *dbIpRacl* | *dbIp6SubIntfAcl* | *dbMplsSystem* | *dbIpTapAgg* | *dbElkResultIpPacl* | *dbIpMcastRouting* | *swEgressIpBridge* | *dbIpSrcXformed* | *dbIpUcastRouting* | *dbEgressMacAcl* | *dbTcCounters* | *dbNone* | *dbIpSubIntfAcl* | *dbVxlanRouting* | *dbElkIp6* | *swInnerIpOGre* | *dbVxlan0* | *dbIpsec* | *dbIpv6MpTt2ndPass* | *dbIpQos* | *dbDirectExtractionPmf1-IntfPolicingCounter* | *dbDirectExtractionPmf3* | *dbMacMirror* | *dbInAc* | *db-VxlanFemDirectExtraction* | *dbDirectExtractionPmf1-StormControlSubIntf* | *dbMplsSubIntfPbr* | *dbEvpnMplsBridging* | *dbIngressEgressCascade* | *dbEgressVxlanDrop* | *dbVniLookup* | *dbElkResultIp6Racl* | *dbEvpnMplsEsiFiltering* | *dbSystemMac* | *dbIp6RaclStage2* | *dbEvpn-MplsEsiLocalToRemote* | *dbTunnelTermination* | *dbIpTunnelIntfPbr* | *dbIp6SubIntfQos* | *dbIpv6VrfDb* | *dbMacTapAgg* | *dbDirect-ExtractionPmf3Srv6* | *dbIp6Mirror* | *dbEgressIp6Racl* | *dbFemDirectExtraction* | *dbIp6Pbr* | *dbRecircEgressTrap* | *dbIp6Qos* | *dbVxlanMlag* | *dbIp6Pacl* | *dbDirectExtractionSetEcn* | *dbDirectExtractionPmf1-Srv6* | *dbMacQos* | *swIpOMpls* | *dbIpv4TtBasic* | *dbTapPort* | *dbMplsFrr* |	PMF database.

lookupType	*enum: tcam* | *largeExactMatch* | *smallExactMatch* | *direct* | *directExtraction* | *doubleWideTcam*	PMF key lookup type.

2.1238.1.1.2 Preselector

context	*integer*	Context ID.
databases	*list of PmfDatabase* see 2.1238.1.1.2.1	List of PMF databases.
line	*integer*	Selection CAM line.
preselectorFields	*table(string) = PreselectorField* see 2.1238.1.1.2.2	Mapping from preselector field name to value and mask.

2.1238.1.1.2.1 PmfDatabase

db	*enum: dbSystem6* | *dbCbf* | *dbDirectExtractionSet-UnderlaySysInLif* | *dbTrafficPolicyIp6* | *dbDirectExtractionPmf1-GroupPolicer* | *dbDummy* | *dbEgressIpAcl* | *dbTrafficPolicyIp4Egress* | *dbIpCpuTrafficPolicy* | *dbL4SrcPortXformed* | *dbIpRacl* | *dbIp6SubIntfAcl* | *dbMplsSystem* | *dbIpTapAgg* | *dbElkResultIpPacl* | *dbIpMcastRouting* | *swEgressIpBridge* | *dbIpSrcXformed* | *dbIpUcastRouting* | *dbEgressMacAcl* | *dbTcCounters* | *dbNone* | *dbIpSubIntfAcl* | *dbVxlanRouting* | *dbElkIp6* | *swInnerIpOGre* | *dbVxlan0* | *dbIpsec* | *dbIpv6MpTt2ndPass* | *dbIpQos* | *dbDirectExtractionPmf1-IntfPolicingCounter* | *dbDirectExtractionPmf3* | *dbMacMirror* | *dbInAc* | *db-VxlanFemDirectExtraction* | *dbDirectExtractionPmf1-StormControlSubIntf* | *dbMplsSubIntfPbr* | *dbEvpnMplsBridging* | *dbIngressEgressCascade* | *dbEgressVxlanDrop* | *dbVniLookup* | *dbElkResultIp6Racl* | *dbEvpnMplsEsiFiltering* | *dbSystemMac* | *dbIp6RaclStage2* | *dbEvpn-MplsEsiLocalToRemote* | *dbTunnelTermination* | *dbIpTunnelIntfPbr* | *dbIp6SubIntfQos* | *dbIpv6VrfDb* | *dbMacTapAgg* | *dbDirect-ExtractionPmf3Srv6* | *dbIp6Mirror* | *dbEgressIp6Racl* | *dbFemDirectExtraction* | *dbIp6Pbr* | *dbRecircEgressTrap* | *dbIp6Qos* | *dbVxlanMlag* | *dbIp6Pacl* | *dbDirectExtractionSetEcn* | *dbDirectExtractionPmf1-Srv6* | *dbMacQos* | *swIpOMpls* | *dbIpv4TtBasic* | *dbTapPort* | *dbMplsFrr* |	PMF database.

2.1238.1.1.2.2 PreselectorField

mask	*integer*	Field mask.
value	*integer*	Field value.

2.1239 show platform (fap | arad | jericho | jericho2) pmf dbinfo [...]

faps	*table(string) = TcamDbInfoTable* see 2.1239.1	TCAM feature DbInfo information, keyed by FAP name.

2.1239.1 TcamDbInfoTable

dbActions	*table(string) = TcamDbInfoActionTable* see 2.1239.1.1	Database actions, keyed by DB name.
dbInfoDatabases	*table(string) = TcamDbInfoData* see 2.1239.1.2	DbInfo databases, keyed by DB name.
dbKeys	*table(string) = TcamDbInfoKeyTable* see 2.1239.1.3	Database keys, keyed by DB name.
dbQualifiers	*table(string) = TcamDbInfoQualifierTable* see 2.1239.1.4	Database qualifiers, keyed by DB name.

2.1239.1.1 TcamDbInfoActionTable

ruleActions	*table(string) = TcamDbInfoActionData* see 2.1239.1.1.1	Rule actions, keyed by rule name.

2.1239.1.1.1 TcamDbInfoActionData

actionLsb	*integer*	Action Lsb.
actionMsb	*integer*	Action Msb.
isFem	*boolean*	This is Fem.
sdkAction	*boolean*	This is an Sdk Action.

2.1239.1.2 TcamDbInfoData

andBits	*integer*	And Bits.
keySize	*integer*	Key Size in bits.
orBits	*integer*	Or Bits.
ownerBits	*enum: Flp | Erpp | Pmf | Vtt*	Owner bits.
[primaryDb]	*string*	Primary DB.
profileIndex	*integer*	Profile Index.
[reservedBanks]	*list of integer*	Reserved Banks.

2.1239.1.3 TcamDbInfoKeyTable

ruleKeys	*table(numeric string) = TcamDbInfoKeyData* see 2.1239.1.3.1	Rule keys, keyed by rule name.

2.1239.1.3.1 TcamDbInfoKeyData

cycle	*integer*	cycle.
key	*string*	Key.

2.1239.1.4 TcamDbInfoQualifierTable

ruleQualifiers	*table(string) = TcamDbInfoQualifierData* see 2.1239.1.4.1	Rule Qualifiers, keyed by rule name.

2.1239.1.4.1 TcamDbInfoQualifierData

keyLsb	*integer*	Key Lsb.
keyMsb	*integer*	Key Msb.
keyOffset	*integer*	Key Offset.
lostBits	*integer*	Lost Bits.
nCopyBits	*integer*	Number Copy Bits.
qualLsb	*integer*	Qualifer Lsb.
qualMsb	*integer*	Qualifer Msb.
second	*boolean*	This qualifier is on the second key.

2.1240 show platform (fap | arad | jericho | jericho2) ppmc multicast [...]

[cmdErr]	*string*	Error Message.
[ppmcEntries]	*table(numeric string) = PpmcEntries* see 2.1240.1	A collection of the mappings from ppmc index to (global outlif0, global outlif1) keyed by fapId.

2.1240.1 PpmcEntries

entries	*list of PpmcDoubleOutlifEntry* see 2.1240.1.1	List of ppmc multicast entries per fap.

2.1240.1.1 PpmcDoubleOutlifEntry

globalOutlif0	*integer*	Global Outlif0.
globalOutlif1	*integer*	Global Outlif1.
index	*integer*	Ppmc Index.

2.1241 show platform (fap | arad | jericho | jericho2) priority-flow-control buffer profile [...]

faps	*table(string) = FapConfig* see 2.1241.1	Mapping of FAP name to its PFC thresholds.
globalConfig	*GlobalConfig* see 2.1241.2	All global CLI config parameters related to PFC.

2.1241.1 FapConfig

rateClasses	*table(string) = Threshold* see 2.1241.1.1	A mapping of rate class to its thresholds.

2.1241.1.1 Threshold

flowControl	*FlowControlThreshold* see 2.1241.1.1.1	Flow control threshold parameters.
reject	*RejectThreshold* see 2.1241.1.1.2	Reject threshold parameters.

2.1241.1.1.1 FlowControlThreshold

alpha	*integer*	FADT alpha.
maxThreshold	*integer*	FADT max flow control threshold in bytes.
minThreshold	*integer*	FADT min flow control threshold in bytes.
offset	*integer*	Resume offset in bytes.

2.1241.1.1.2 RejectThreshold

alpha	*integer*	FADT alpha.
maxThreshold	*integer*	FADT max reject threshold in bytes.
minThreshold	*integer*	FADT min reject threshold in bytes.

2.1241.2 GlobalConfig

bufferUtilizationFactor	*integer*	DRAM utilization factor in percentage used in determining PFC thresholds in bytes from the number of buffers available for each source port and traffic class.
defaultBufferUtilization-Factor	*integer*	Default DRAM utilization factor in percentage.
defaultNumTrafficClasses	*integer*	Default number of traffic classes in use.
defaultXoffTime	*integer*	Default XOFF duration in microseconds.
defaultXonTime	*integer*	Default XON duration in microseconds.
numTrafficClasses	*integer*	Number of traffic classes in use.
xoffTime	*integer*	Configured XOFF duration in microseconds.
xonTime	*integer*	Configured XON duration in microseconds.

2.1242 show platform (fap | arad | jericho | jericho2) ptp hardware-sync [...]

slaves	*table(string) = SlaveModel* see 2.1242.1	A Mapping between a TimeSync Slave and its time synchronization information.

2.1242.1 SlaveModel

[crcErrors]	*integer*	CRC Error(s).
[fifoRetryLimitExceeded] **	*integer*	FIFO Retry Limit Exceeded Error(s). **Since: 4.18.1
[invalidTimestampErrors] **	*integer*	Invalid Timestamp Error(s). **Since: 4.18.1
[localCounter]	*integer*	Latest Local Counter.
[masterTod]	*integer*	Latest Master Time of Day (ToD).
[phaseDifferences]	*list of integer*	Last 4 phase differences (in ns).
[retriesLeftSelfTest] **	*integer*	Retries Left after self test. **Since: 4.21.3
[slaveFreqErrors] **	*integer*	Slave Frequency Error(s). **Since: 4.18.1
state	*enum: stable | switchover | disabled | initialized | synchronizing | failed*	Operational State.
[timeToStability]	*integer*	Time to stability (in s).
[timeoutErrors] **	*integer*	Timeout Error(s). **Since: 4.18.1
[updates]	*integer*	Sync Update(s).

2.1243 show platform (fap | arad | jericho | jericho2) qos tcam mapping [...]

fapTcamMapping	*table(string) = FapQosAclTcamMappingModel* see 2.1243.1	List of tcam mapping models per fap.

2.1243.1 FapQosAclTcamMappingModel

fapName	*string*	Name of the fap.
pmapMapping	*table(string) = PolicyMapTcamMappingModel* see 2.1243.1.1	Policy-map models on the fap.

2.1243.1.1 PolicyMapTcamMappingModel

policyMapName	*string*	Name of policy-map.
policyTypeMapping	*table(string) = PolicyTypeTcamMappingModel* see 2.1243.1.1.1	Mapping models per policy type.

2.1243.1.1.1 PolicyTypeTcamMappingModel

policyKeyTypeMapping	*table(string) = PolicyKeyTypeTcamMappingModel* see 2.1243.1.1.1.1	Mapping models per policy key type.
policyType	*string*	Type of class-maps listed in the model.

2.1243.1.1.1.1 PolicyKeyTypeTcamMappingModel

appliedTo	*integer*	Applied to.
classMapMapping	*table(string) = ClassMapTcamMappingModel* see 2.1243.1.1.1.2	Class-maps in the policy-map.
classMapOrder	*list of string*	Order of class-maps in policy-map.
dbType	*string*	Db used by the class-maps listed in the model.
keyType	*string*	Key Type.

2.1243.1.1.1.2 ClassMapTcamMappingModel

classMapName	*string*	Class-map name.
[classOffset] **	*string*	Associated class-offset, if configured. **Since: 4.28.2
[counterIndex]	*string*	Associated counterB index, if configured.
[policerIndex]	*string*	Associated policer-meter index, if configured.
ruleRange	*RuleRangeModel* see 2.1243.1.1.1.3	Associated range of rules in aad.

tcamRange	*TcamRangeModel* see 2.1243.1.1.1.4	Associated range of tcam-entries.

2.1243.1.1.1.3 RuleRangeModel

end	*integer*	End of range.
start	*integer*	Start of range.

2.1243.1.1.1.4 TcamRangeModel

discontiguousRange **	*boolean*	TCAM-Range is discontiguous. **Since: 4.23.1
endEntry	*TcamEntryModel* see 2.1243.1.1.1.5	End entry of tcam range.
startEntry	*TcamEntryModel* see 2.1243.1.1.1.5	Start entry of tcam range.

2.1243.1.1.1.5 TcamEntryModel

banks	*list of integer*	Bank in which the tcam entry is located.
offset	*integer*	Offset of the entry in bank.

2.1244 show platform (fap | arad | jericho | jericho2) qos tcam mapping bank BANK ENTRY [...]

fapTcamEntry	*table(string) = FapQosAclTcamEntryModel* see 2.1244.1	TcamEntry Model per fap.

2.1244.1 FapQosAclTcamEntryModel

[aclGroup]	*string*	Corresponding aclGroup.
[aclNumVal]	*integer*	aclNum.
[aclType]	*string*	aclType.
[aclTypes]	*integer*	Number of aclTypes.
actions	*list of ActionModel* see 2.1244.1.2	dpMeterCommand action.
[classMap]	*string*	Name of class-map associated to the entry.
[dbType]	*string*	dbType.
[fapName]	*string*	Name of the fap.
[matchingRuleId]	*integer*	Corresponding rule-id in the associated aad.
[policyMap]	*string*	Name of policy-map associated to the entry.
qosEntry	*boolean*	Does Qos feature own this entry?
qualifiers	*list of QualifierModel* see 2.1244.1.1	List of qualifiers in the tcam-entry.
[skippedEntry]	*boolean*	Is it a skipped entry?
[startRuleId]	*integer*	First rule id of the associated class-map.

2.1244.1.1 QualifierModel

mask	*integer*	Qualifier mask.
name	*string*	Qualifier name.
value	*integer*	Qualifier value.

2.1244.1.2 ActionModel

fields **	*table(string) = integer*	Action field values, keyed by action field names. **Since: 4.23.2
name	*string*	Action name.
[valid]	*boolean*	Is action valid?
value	*integer*	Action value.

2.1245 show platform (fap | arad | jericho | jericho2) qos tcam mapping policy-map PMAP [...]

see 2.1243

2.1246 show platform (fap | arad | jericho | jericho2) sflow accel [...]

collectorVrf	*string*	sFlow accel selected collector VRF.
fapResourceInfo	*table(string) = PlatformSflowAccelResourceInfo* see 2.1246.1	sFlow accel resources keyed by fapName.

2.1246.1 PlatformSflowAccelResourceInfo

mcid	*integer*	sFlow accel multicast Id.
outLif	*integer*	sFlow accel global outLif.
tunnelOutLifs	*list of PlatformSflowAccelTunnelOutlifInfo* see 2.1246.1.1	sFlow accel tunnel outLifs for this collector.

2.1246.1.1 PlatformSflowAccelTunnelOutlifInfo

collectorIP	*IP address*	source address for group interface.
tunnelOutlif	*integer*	sFlow accel global outlif.
udpPort	*integer*	sFlow accel UDP port associated with collector IP.
vlanId	*integer*	Internal VLAN Id mapped to this collector VRF.

2.1247 show platform (fap | arad | jericho | jericho2) sflow accel counters [...]

countersDict	*table(string) = PlatformSflowAccelCountersInfo* see 2.1247.1	sFlow accel counters keyed by str(fapName.coreId).

2.1247.1 PlatformSflowAccelCountersInfo

intfCounterList	*list of PlatformSflowAccelIntfCounters* see 2.1247.1.1	sFlow accel counters interface list for specific fap and core.

2.1247.1.1 PlatformSflowAccelIntfCounters

counterIdx	*integer*	Counter Index.
counterValue	*integer*	Counter Value.
interfaceName	*interface name*	Interface Name.

2.1248 show platform (fap | arad | jericho | jericho2) subinterface mapping [...]

subInterfaces	*table(string) = SubInterfaceMap* see 2.1248.1	A mapping of subinterface name to its hardware programming details.

2.1248.1 SubInterfaceMap

parentInterfaces	*table(string) = SubIntfInfo* see 2.1248.1.1	A mapping of parent interface name to its subinterface details.

2.1248.1.1 SubIntfInfo

eveProfile **	*integer*	Egress vlan edit profile index. **Since: 4.29.1
fap	*integer*	FAP ID.
fapPortId	*integer*	FAP port ID.
lifEntryVsi	*integer*	Virtual switch instance.
lifIndex	*integer*	Logical interface index.
outAcIndex	*integer*	Output attachment circuit index.
outAcTag	*string*	Output attachment circuit tag.
vlanDomain	*integer*	VLAN domain.

2.1249 show platform (fap | arad | jericho | jericho2) subinterface mapping l2 [...]

subInterfaces	*table(Interface name) = SubInterfaceL2Map* see 2.1249.1	A mapping of subinterface name to its hardware programming details.

2.1249.1 SubInterfaceL2Map

parentInterfaces	*table(Interface name) = SubIntfL2Info* see 2.1249.1.1	A mapping of parent interface name to its subinterface details.

2.1249.1.1 SubIntfL2Info

encapVlanTag	*string*	Output attachment circuit tag.
eveProfile **	*integer*	Egress vlan edit profile index. **Since: 4.29.1
fap	*integer*	FAP ID.
fapPortId	*integer*	FAP port ID.
glemIndex	*integer*	GlobalOutlif.
lifEntryVsi	*integer*	Virtual switch instance.
lifIndex	*integer*	Logical interface index.
outAcIndex	*integer*	Output attachment circuit index.
vlanDomain	*integer*	VLAN domain.

2.1250 show platform (fap | arad | jericho | jericho2) subinterface mapping l3 [...]

subInterfaces	*table(string) = SubInterfaceL3Map* see 2.1250.1	A mapping of subinterface name to its hardware programming details.

2.1250.1 SubInterfaceL3Map

parentInterfaces	*table(string) = SubIntfL3Info* see 2.1250.1.1	A mapping of parent interface name to its subinterface details.

2.1250.1.1 SubIntfL3Info

encapVlanTag	*string*	Output attachment circuit tag.
eveProfile **	*integer*	Egress vlan edit profile index. **Since: 4.29.1
fap	*integer*	FAP ID.
fapPortId	*integer*	FAP port ID.
lifEntryVsi	*integer*	Virtual switch instance.
lifIndex	*integer*	Logical interface index.
outAcIndex	*integer*	Output attachment circuit index.
propagationProfile	*integer*	Propagation profile.
protectionPointer	*string*	Protection pointer.
qosProfile	*integer*	QoS profile.
vlanDomain	*integer*	VLAN domain.

2.1251 show platform (fap | arad | jericho | jericho2) tcam action-priority [...]

faps	*table(string) = TcamFapActionPriorityInfo* see 2.1251.1	TCAM feature action priority information, keyed by FAP name.

2.1251.1 TcamFapActionPriorityInfo

databases	*table(string) = TcamActionPriorityInfo* see 2.1251.1.1	TCAM action priorities, keyed by db name.

2.1251.1.1 TcamActionPriorityInfo

dbActionPriorityActive	*integer*	Active TCAM database action priority.
dbActionPriorityDefault	*integer*	Default TCAM database action priority.

2.1252 show platform (fap | arad | jericho | jericho2) tcam summary [...]

faps	*table(string) = TcamFapInfo* see 2.1252.1	Map of FAPs to their TCAM bank information.

2.1252.1 TcamFapInfo

bankInfo	*list of TcamBankInfo* see 2.1252.1.1	List of TCAM bank numbers and their corresponding TCAM databases.

2.1252.1.1 TcamBankInfo

bankNum	*integer*	TCAM bank number.
dbName	*string*	TCAM database used name.
dbNameReserved **	*string*	TCAM database reserved name. **Since: 4.20.5

2.1253 show platform (fap | arad | jericho | jericho2) vpls pseudowire decap [...]

revision 2

pseudowires	*table(Interface name) = VplsPseudowireDetails* see 2.1253.1	VPLS decap entry information keyed by pw interface.

2.1253.1 VplsPseudowireDetails

active	*boolean*	Pseudowire is active.
[controlWord]	*boolean*	ControlWord is present.
[flowLabel]	*boolean*	FlowLabel is present.
[globalInLif]	*integer*	GlobalInLif value.
[hwFec]	*integer*	Hardware FEC table index.
[inLifProfile]	*integer*	Jericho2LifProfiles value.
[instanceContext]	*string*	VPLS-pwInstance-neighbour IP-pwId-pwInterface.
label	*integer*	VPLS MPLS label.
[orientation]	*integer*	In/Out Orinetation value.
[pseudowireGroup]	*string*	Pw group name.
termProfile	*integer*	Index to Tunnel termination profile.
vlans	*table(numeric string) = VlanToTag* see 2.1253.1.1	VPLS Pw information keyed by instfDisplayContext.

2.1253.1.1 VlanToTag

lifIndex	*integer*	Index to inLif table.
[normalizedTag]	*integer*	Dot1Q Tag.

2.1254 show platform (fap | arad | jericho | jericho2) vrf selection inlif [...]

banks	*table(numeric string) = VrfSelectionInLifBank* see 2.1254.1	VRF selection InLIF banks, keyed by index.

2.1254.1 VrfSelectionInLifBank

inLifs	*table(numeric string) = VrfSelectionInLifEntry* see 2.1254.1.1	VRF selection InLIF entries, keyed by index.

2.1254.1.1 VrfSelectionInLifEntry

fallbackVrf	*integer*	Fallback VRF ID.
inLifProfile	*integer*	InLIF profile index.
lemVrf	*integer*	LEM VRF ID.
teVrf	*integer*	TE VRF ID.

2.1255 show platform (fap | arad | jericho | jericho2) vrf selection tcam shadow [...]

[ipv4TcamEntries]	*list of VrfSelectionV4TcamEntries* see 2.1255.2	A collection of IPv4 tcam entries.
[ipv6ExtendedTcamEntries] **	*list of VrfSelection-ExtendedV6TcamEntries* see 2.1255.1	A collection of IPv6 tcam entries. **Since: 4.30.1
[ipv6TcamEntries]	*list of VrfSelectionV6TcamEntries* see 2.1255.3	A collection of IPv6 tcam entries.

2.1255.1 VrfSelectionExtendedV6TcamEntries

bank	*string*	TCAM Bank.
dscp	*integer*	DSCP.
dscpMask	*integer*	DSCP Mask.
index	*integer*	TCAM Index.
label	*integer*	Policy Label.
labelMask	*integer*	Policy Label Mask.
lemVrf	*integer*	LEM VRF ID.
sourcePrefix	*IP address prefix*	IPv6 Source prefix.
teVrf	*integer*	TE VRF ID.

2.1255.2 VrfSelectionV4TcamEntries

bank	*integer*	TCAM Bank.
dscp	*integer*	DSCP.
dscpMask	*integer*	DSCP Mask.
inLifBank	*integer*	InLif Bank.
inLifIndex	*integer*	InLif Index.
index	*integer*	TCAM Index.
ipProtocol	*integer*	IP Protocol Mask.
ipProtocolMask	*integer*	IP Protocol.
label	*integer*	Policy Label.
labelMask	*integer*	Policy Label Mask.
sourcePrefix **	*IP address prefix*	IP Source prefix. **Since: 4.27.0

2.1255.3 VrfSelectionV6TcamEntries

bank	*integer*	TCAM Bank.
dscp	*integer*	DSCP.
dscpMask	*integer*	DSCP Mask.
inLifBank	*integer*	InLif Bank.
inLifIndex	*integer*	InLif Index.
index	*integer*	TCAM Index.
label	*integer*	Policy Label.

labelMask	*integer*	Policy Label Mask.

2.1256 show platform (fap | arad | jericho | jericho2) vxlan multicast route underlay [...]

underlayMrouteEntries	*list of VxlanUnderlayMrouteEntry* see 2.1256.1	A collection of underlay multicast routes with EVPN VxLAN.

2.1256.1 VxlanUnderlayMrouteEntry

destination	*IP address*	Destination IP address.
lifIndex	*integer*	Logical interface index.
source	*IP address*	Source IP address.
vlanId	*integer*	VLAN ID.

2.1257 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric [...]

fapFeSerdesInfo	*table(string) = FapFeSerdesInfo* see 2.1257.1	Maps Fap/Fe name to all its serdes.

2.1257.1 FapFeSerdesInfo

linkActiveMask	*list of integer*	Link Active Mask.
peerSerdesInfo	*table(numeric string) = SerdesInfo* see 2.1257.1.1	Maps SerdesId to peer's Serdes information.
serdesInfo	*table(numeric string) = SerdesInfo* see 2.1257.1.1	Maps SerdesId to Serdes information.

2.1257.1.1 SerdesInfo

adminStatus	*enum: Up | Down*	Serdes Admin status.
[correctedCells]	*integer*	Corrected cells.
[correctedCellsLastOccurred] **	*floating point number*	Time Corrected cells last occurred in UTC. **Since: 4.17.0
[correctedCellsOverflow]	*boolean*	Corrected cells overflowed.
crcErrorCells	*integer*	Cells received with CRC errors.
crcErrorCellsOverflow	*boolean*	Cells received with CRC errors overflowed.
[crcErrorsLastOccurred]	*floating point number*	Time CRC errors last occurred in UTC.
[fecCells]	*integer*	FEC corrected cells.
[fecCellsLastOccurred]	*floating point number*	Time FEC cells last occurred in UTC.
[fecCellsOverflow]	*boolean*	FEC corrected cells.
interruptsInfo	*table(string) = InterruptInfo* see 2.1257.1.1.1	Maps Interrupt name to interrupt info.
isolatedCount	*integer*	Operational status changed to Isolated count.
[isolatedLastOccurred]	*floating point number*	Time status change to Isolated occurred in UTC.
leakyBucketTokens	*integer*	Tokens in Leaky Bucket which monitors serdes health.
maxInFifoCells	*integer*	Maximum cells In-Fifo.
[maxInFifoLastOccurred]	*floating point number*	Time maximum In-Fifo last occurred in UTC.
maxOutFifoCells	*integer*	Maximum cells in Out-Fifo.
[maxOutFifoLastOccurred]	*floating point number*	Time maximum Out-Fifo last occurred in UTC.
operationalStatus	*enum: Cleared | Isolated | Up | Down*	Serdes Operational status.
[peerFapFeName]	*string*	Peer's Fap/Fe name.
[peerSerdesId]	*integer*	Peer's Serdes Id.
[rxCells]	*integer*	Received cells.
[rxCellsOverflow]	*boolean*	Received cells overflowed.

[rxControlCells]	*integer*	Received control cells.
[rxControlCellsOverflow]	*boolean*	Received control cells overflowed.
[rxDataRate] **	*integer*	Receive data rate in Mbps. **Since: 4.21.3
[txCells]	*integer*	Transmitted cells.
[txCellsOverflow]	*boolean*	Transmitted cells overflowed.
[txDataRate] **	*integer*	Transmit data rate in Mbps. **Since: 4.21.3

2.1257.1.1.1 InterruptInfo

interruptCount	*integer*	Interrupt occurred count.
[lastOccurred]	*floating point number*	Time interrupt last occurred in UTC.

2.1258 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric detail [...]

see 2.1257

2.1259 show platform (fap | fe | arad | jericho | jericho2 | fe1600 | fe3600 | ramon) fabric rates mbps [...]

see 2.1257

2.1260 show platform (fap | jericho2) mac security [...]

faps	*table(string) = MacSecTableFap* see 2.1260.1	A mapping of FAP name to MACsec properties.

2.1260.1 MacSecTableFap

forwardingEngineEncryption	*boolean*	Forwarding engine encryption is enabled.
[interfaces]	*table(string) = InterfacesEntry* see 2.1260.1.1	A mapping of interface name to MACsec properties.
is2bAssocNumSupported	*boolean*	2b association number is supported.

2.1260.1.1 InterfacesEntry

[bypass]	*list of string*	Bypass MACsec encryption/decryption.
[egress]	*EgressEntry* see 2.1260.1.1.1	The egress indicies.
enabled	*boolean*	MACsec is enabled.
[ingress]	*IngressEntry* see 2.1260.1.1.2	The ingress indicies.
instance	*integer*	The MACsec instance #.
port	*integer*	The MACsec port #.
[svTag]	*integer*	The svTag #.

2.1260.1.1.1 EgressEntry

saIndex	*integer*	The egress saIndex.
scIndex	*integer*	The egress scIndex.

2.1260.1.1.2 IngressEntry

saIndex	*list of integer*	The ingress saIndex.
scIndex	*integer*	The ingress scIndex.
spIndex	*integer*	The ingress spIndex.

2.1261 show platform (fe | fe1600 | fe3600 | ramon) [...]

switches	*table(string) = FeInfo* see 2.1261.1	Maps fabric chip names to their corresponding chip information.

2.1261.1 FeInfo

[feId]	*integer*	Fabric chip ID.
[feRevisionMajor] **	*string*	FE major revision. **Since: 4.30.1

`[feRevisionMinor]` **	*string*	FE minor revision. **Since: 4.30.1
`[feType]` **	*string*	Chip type name. **Since: 4.30.1

2.1262 show platform (sand | arad) lag mode

`configLagMode`	*enum: 512x32 | 2048x8 | None | 64x256 | 256x64 | 1024x16 | 128x128*	Configured LAG mode.
`statusLagMode`	*enum: 512x32 | 2048x8 | None | 64x256 | 256x64 | 1024x16 | 128x128*	Actual LAG mode.

2.1263 show platform agent-state verify (AGENT | all)

`agentVerifyState`	*table(string) = PerAgentVerifyResultDict* see 2.1263.1	Agent verify state indexed by agent Name.

2.1263.1 PerAgentVerifyResultDict

`perAgentResult`	*table(string) = string*	dict of { sm name : error str }.

2.1264 show platform barefoot acl

`chips`	*table(string) = ProgrammingStatusEntry* see 2.1264.1	Chip programming status.

2.1264.1 ProgrammingStatusEntry

`dirty`	*list of string*	List of dirty tables.
`globalBlock`	*enum: blockRemove | blockApply | blockOn | blockOff*	Global Acl Block status.
`smMode`	*enum: program | idle | removeBlock | applyBlock*	Programming phase.

2.1265 show platform barefoot acl crlid

`chips`	*table(string) = CrlIdSummaryEntry* see 2.1265.1	Chip crlId status.

2.1265.1 CrlIdSummaryEntry

activeV4Intf	*integer*	number of active v4 intf.
activeV4SubIntfVlan	*integer*	number of active v4 subintf.
activeV4Vlan	*integer*	number of active v4 vlan.
activeV6Intf	*integer*	number of active v6 intf.
activeV6SubIntfVlan	*integer*	number of active v6 subintf.
activeV6Vlan	*integer*	number of active v6 vlan.
crlIdMode	*enum: crlIdBlockOff | crlIdBlockOn*	CrlId block status.

2.1266 show platform barefoot acl crlid active interface

chips	*table(string) = CrlIdChipActiveIntf* see 2.1266.1	crlId active intf status.

2.1266.1 CrlIdChipActiveIntf

entries	*table(string) = CrlIdActiveIntfEntry* see 2.1266.1.1	Chip crlId active intf status.

2.1266.1.1 CrlIdActiveIntfEntry

v4CrlId	*integer*	v4 ACL CrlId.
v6CrlId	*integer*	v6 ACL CrlId.

2.1267 show platform barefoot acl crlid active subinterface

chips	*table(string) = CrlIdChipActiveSubIntf* see 2.1267.1	crlId active subintf status.

2.1267.1 CrlIdChipActiveSubIntf

entries	*table(string) = CrlIdActiveSubIntfEntry* see 2.1267.1.1	Chip crlId active subintf status.

2.1267.1.1 CrlIdActiveSubIntfEntry

v4CrlId	*integer*	v4 CrlId.
v4VlanId	*integer*	v4 internal vlan.
v6CrlId	*integer*	v6 CrlId.
v6VlanId	*integer*	v6 internal vlan.

2.1268 show platform barefoot acl crlid active vlan

chips	*table(string) = CrlIdChipActiveVlan* see 2.1268.1	crlId active vlan status.

2.1268.1 CrlIdChipActiveVlan

entries	*table(numeric string) = CrlIdActiveVlanEntry* see 2.1268.1.1	Chip crlId active vlan status.

2.1268.1.1 CrlIdActiveVlanEntry

v4CrlId	*integer*	v4 ACL CrlId.
v6CrlId	*integer*	v6 ACL CrlId.

2.1269 show platform barefoot acl crlid table racl

chips	*table(string) = CrlIdChipTableRacl* see 2.1269.1	crlId active subintf status.

2.1269.1 CrlIdChipTableRacl

entries	*list of CrlIdTableRaclEntry* see 2.1269.1.1	Chip crlId table racl status.

2.1269.1.1 CrlIdTableRaclEntry

subIntfCrlId	*integer*	subIntf CrlId.
sviCrlId	*integer*	svi CrlId.
tableType	*enum: ipv4RAcl | ipv6RAcl*	racl type.
vlanId	*integer*	internal vlan.

2.1270 show platform barefoot acl l4port

chips	*table(string) = L4PortChipSummary* see 2.1270.1	L4Port status.

2.1270.1 L4PortChipSummary

entries	*table(string) = L4PortEntry* see 2.1270.1.1	Per-Chip prefix status.

2.1270.1.1 L4PortEntry

activeEncodings	*integer*	Number of active encodings.
activeInternalId	*floating point number*	Active Request Internal Id.
activeRequestId	*floating point number*	Active Request Id.
activeXforms	*integer*	Number of active xforms.
attemptedEncodings	*integer*	Number of attempted encodings.
attemptedInternalId	*floating point number*	Attempted Request Internal Id.
attemptedRequestId	*floating point number*	Attempted Request Id.
attemptedResponseId	*floating point number*	Attempted Response Id.
attemptedSuccess	*boolean*	Attempted success.
attemptedXforms	*integer*	Number of attempted xforms.

2.1271 show platform barefoot acl prefix

chips	*table(string) = PrefixChipSummary* see 2.1271.1	Prefix status.

2.1271.1 PrefixChipSummary

entries	*table(string) = PrefixEntry* see 2.1271.1.1	Per-Chip prefix status.

2.1271.1.1 PrefixEntry

activeNum	*integer*	Number of active xforms.
activeRequestId	*floating point number*	Active Request Id.
activeResponseId	*floating point number*	Active Response Id.
activeSuccess	*boolean*	Active success.
attemptedNum	*integer*	Number of attempted xforms.
attemptedRequestId	*floating point number*	Attempted Request Id.
attemptedResponseId	*floating point number*	Attempted Response Id.
attemptedSuccess	*boolean*	Attempted success.
rollback **	*boolean*	Attempted Rollback status. **Since: 4.21.3

2.1272 show platform barefoot agent restarts

chipRestarts	*table(string) = BfnChipRestartsPerSourceModel* see 2.1272.1	Each restart requester's restart requests.

2.1272.1 BfnChipRestartsPerSourceModel

chipRestartsByTime	*table(string) = BfnChipRestartEntry* see 2.1272.1.1	This restart source's restart requests.
restartSource	*string*	Source of restart.

2.1272.1.1 BfnChipRestartEntry

chipName	*string*	Chip that was restarted.
restartCause	*string*	Cause of restart.
restartReason	*string*	Reason of restart.
restartSource	*string*	Source of restart.
restartTime	*floating point number*	UTC Time restart was requested.
restartType	*enum: systemReset | featureAgentReset | chipReset | sliceReset | systemWithFeatureReset*	Type of restart.

2.1273 show platform barefoot bfd-offload intervals

txIntervalA	*integer*	TX interval for BFD offloaded sessions in units of milliseconds.
txIntervalB	*integer*	TX interval for BFD offloaded sessions in units of milliseconds.

2.1274 show platform barefoot bfd-offload session DISCRIMINATOR | summary

offloadSessions	*list of BfnBfdOffloadSession* see 2.1274.1	Collection of BFD offloaded sessions.

2.1274.1 BfnBfdOffloadSession

discriminator	*integer*	Session discrimiator.
[lastRxTimeMs]	*integer*	Number of milliseconds since last packet was received.
profileId	*integer*	Profile ID.
[rxPkts]	*integer*	Number of received BFD packets.
rxSessionId	*integer*	RX Session ID.
[txPkts]	*integer*	Number of sent BFD packets.
txSessionId	*integer*	TX Session ID.

2.1275 show platform barefoot counters nat interfaces [...]

interfaces	*table(Interface name) = InterfacesCountersNatModel* see 2.1275.1	Mapping between an interface and NAT counters.

2.1275.1 InterfacesCountersNatModel

inSnatOctets	*integer*	Bytes with source NAT.
inSnatPkts	*integer*	Packets with source NAT.

2.1276 show platform barefoot counters table [...]

tableCounters	*list of PipeTableCounter* see 2.1276.1	List of pipe table counters.

2.1276.1 PipeTableCounter

deviceId	*integer*	Device Id of chip.
entries	*table(numeric string) = TableCounterEntry* see 2.1276.1.1	Dict of the identifiers and its counts in a given pipe. The identifier is the entry handle for direct tables, and counter index for indirect tables.
isBytes	*boolean*	Count bytes.
isDirect	*boolean*	Direct or indirect table.
isPackets	*boolean*	Count packets.
name	*string*	Name of table counter.
pipeId	*integer*	Specific pipe for a table counter.

2.1276.1.1 TableCounterEntry

bytesCount	*integer*	Count of bytes.
identifier	*string*	Entry identifier - counter index or match spec.
packetsCount	*integer*	Count of packets.

2.1277 show platform barefoot int drop codes

dropReasonCodes	*table(numeric string) = DropReasonCodeEntry* see 2.1277.1	Mapping for drop codes to reasons.

2.1277.1 DropReasonCodeEntry

dropReason	*string*	Drop reason.

2.1278 show platform barefoot l2 mac address-table

vlans	*table(numeric string) = VlanMacAddressTable* see 2.1278.1	Dictionary mapping a Vlan ID to a set of MAC addresses learnt or configured on that Vlan.

2.1278.1 VlanMacAddressTable

macAddresses	*table(MAC address) = MacTableEntry* see 2.1278.1.1	Dictionary mapping a MAC address to the corresponding outgoing interface.

2.1278.1.1 MacTableEntry

addressType	*enum: drop* | *default* | *dynamic* | *static*	Type of MAC address.
l2Aintf	*integer*	Layer-2 Arista Interface ID corresponding to the destination interface.
port	*interface name*	Destination interface for packets matching this MAC table entry.

2.1279 show platform barefoot l2 mac address-table diff

vlanToMac	*table(numeric string) = VlanMacAddress* see 2.1279.1	Dictionary mapping a VLAN ID to a set of MAC Addresses on that VLAN that have different states in hardware and software.

2.1279.1 VlanMacAddress

macAddressToDiffEntries	*table(MAC address) = DiffMacTableEntry* see 2.1279.1.1	Dictionary mapping MAC Address to its states in hardware and software.

2.1279.1.1 DiffMacTableEntry

[hwEntryType]	*enum: Learned Remote MAC* | *Peer Configured Remote MAC* | *Peer Received Remote MAC* | *Peer Dynamic MAC* | *Peer Learned Remote MAC* | *Learned Dynamic MAC* | *Peer Static MAC* | *Received Remote MAC* | *Configured Remote MAC* | *Configured Static MAC*	MAC Address entry type in hardware.
[hwInterface]	*interface name*	Destination interface of the entry in hardware.
[swEntryType]	*enum: Learned Remote MAC* | *Peer Configured Remote MAC* | *Peer Received Remote MAC* | *Peer Dynamic MAC* | *Peer Learned Remote MAC* | *Learned Dynamic MAC* | *Peer Static MAC* | *Received Remote MAC* | *Configured Remote MAC* | *Configured Static MAC*	MAC Address entry type in software.
[swInterface]	*interface name*	Destination interface of the entry in software.

2.1280 show platform barefoot l2 mac address-table pending

pendingEntries	*list of PendingMacTableEntry* see 2.1280.1	Pending L2 entries which have not been programmed in hardware.

2.1280.1 PendingMacTableEntry

entryType	*enum: Learned Remote MAC | Peer Configured Remote MAC | Peer Received Remote MAC | Peer Dynamic MAC | Peer Learned Remote MAC | Learned Dynamic MAC | Peer Static MAC | Received Remote MAC | Configured Remote MAC | Configured Static MAC*	MAC Address entry type.
intfId	*interface name*	Destination interface matching this MAC Address.
macAddress	*MAC address*	MAC Address.
vlanId	*integer*	VLAN ID of the interface.

2.1281 show platform barefoot l2 stp blocked [...]

vlans	*table(numeric string) = PortsInVlan* see 2.1281.1	Dictionary mapping a Vlan ID to the corresponding member interfaces.

2.1281.1 PortsInVlan

interfaces	*list of Interface name*	List of interfaces that are members of a given vlan.

2.1282 show platform barefoot l2 summary

revision 2

backlogSize	*integer*	Number of MAC entries pending in the backlog queue.
evpnMacs **	*integer*	Total number of EVPN MACs. **Since: 4.22.1
macStats	*table(string) = integer*	MAC statictics.
peerMacs	*integer*	Total number of MAC entries learnt from the MLAG peer.
remoteMacs	*integer*	Total number of Vxlan MACs.
tableSummary	*table(string) = TableSummary* see 2.1282.1	Dictionary mapping an L2 table name to its usage statistics.

totalMacs	*integer*	Total number of MAC entries.

2.1282.1 TableSummary

pendingEntries	*integer*	Number of entries waiting to be programmed in hardware.
totalEntries	*integer*	Total number of entries available in the table.
usedEntries	*integer*	Number of entries used in the table.

2.1283 show platform barefoot l3 (routes-v4 | routes-v6) [...]

[cmdErr]	*string*	Error Message.
[vrfs]	*table(numeric string) = IpRoutes* see 2.1283.1	Mapping of VRF ID to IP routes.

2.1283.1 IpRoutes

[cmdErr]	*string*	Error Message.
[routes]	*table(IPv4 or IPv6 address prefix) = IpRouteEntry* see 2.1283.1.1	Mapping prefix to its nextHop Id.
[vrfId]	*integer*	VRF id.

2.1283.1.1 IpRouteEntry

inHostTable	*string*	prefix is in host table.
nextHopId	*integer*	Nexthop index.

2.1284 show platform barefoot l3 (routes-v4 | routes-v6) detail [...]

[cmdErr]	*string*	Error Message.
[vrfs]	*table(numeric string) = IpRoutesDetail* see 2.1284.1	Mapping of VRF ID to IP routes.

2.1284.1 IpRoutesDetail

[cmdErr]	*string*	Error Message.
[routes]	*table(IPv4 or IPv6 address prefix) = UcastNexthop* see 2.1284.1.1	Mapping prefix to its nextHop Id with details.
[vrfId]	*integer*	VRF id.

2.1284.1.1 UcastNexthop

[cmdErr]	*string*	Error Message.
[ecmpGroupId]	*integer*	ECMP group ID.
[ecmpGroupSize]	*integer*	ECMP group size.
[ecmpRefCount]	*integer*	Unicast reference count of the ECMP group.
[l2AdjIndex]	*integer*	Layer 2 Adjacency Index.
[nexthops]	*table(numeric string) = NexthopEntry* see 2.1284.1.1.1	Mapping of nexthopId to nexthop details.

2.1284.1.1.1 NexthopEntry

egressType	*enum: drop | physical | unknown | flood | cpu | tunnel*	Egress port type.
macAddress	*MAC address*	Unicast Non-ECMP next hop mac address.
nexthopTable	*string*	logical table where nexthop is programmed.
[nexthopTableIndex]	*integer*	logical table index.
nexthopType	*integer*	Unicast Non-ECMP next hop type.
refCount	*integer*	Unicast reference count of the next hop.
[tunnelIndex]	*integer*	tunnel index.
[tunnelNexthop]	*integer*	tunnel next hop.
vlanId	*integer*	Unicast Non-ECMP Vlan id.
[vport]	*integer*	virtual port id.

2.1285 show platform barefoot l3 (vteps-v4 | vteps-v6)

revision 2

[cmdErr]	*string*	Error Message.

[vteps]	*table(IPv4 or IPv6 address) = VtepEntry* see 2.1285.1	Mapping of vtep dip to hw table data.

2.1285.1 VtepEntry

ipv6MappingDataA	*integer*	Tunnel ipv6 mappingData upper 32 bits.
ipv6MappingDataB	*integer*	Tunnel ipv6 mappingData lower 32 bits.
ipv6MappingDataIndex	*integer*	Tunnel ipv6 mappingData index.
tunnelDipDataA	*integer*	Tunnel dip/dip2 table data dipA.
tunnelDipDataB	*integer*	Tunnel dip/dip2 table data dipB.
tunnelDipIndex	*integer*	Tunnel dip/dip2 table index.
tunnelDipSlot	*integer*	Tunnel dip/dip2 table slot.
tunnelId	*integer*	Tunnel id.
vtepDipErr	*string*	Error Message.
vtepDipInHw	*string*	IP address in hardware.

2.1286 show platform barefoot l3 events

lpmCompletionTime	*floating point number*	Time since boot of last LPM update.
v4	*IpVersionEvents* see 2.1286.1	IPv4 specific events.
v6	*IpVersionEvents* see 2.1286.1	IPv6 specific events.

2.1286.1 IpVersionEvents

[defaultVrfCompletionTime] **	*floating point number*	Time since boot of last update to the default VRF table. **Since: 4.22.1
hostCompletionTime	*floating point number*	Time since boot of last update to the host table.
lastRouteUpdateTime	*floating point number*	Time since boot of last route update.
routingDisabledTime	*floating point number*	Time since boot of routing being disabled.
routingEnabledTime	*floating point number*	Time since boot of routing being enabled.

2.1287 show platform barefoot l3 interface

intfs	*table(string) = InterfaceEntry* see 2.1287.1	A mapping interface to its properties.

2.1287.1 InterfaceEntry

ipv4McastRouting	*integer*	Indicates if IPv4 mutlticast routing enabled.
ipv4UcastRouting	*integer*	Indicates if IPv4 unicast routing is enabled.
ipv6McastRouting	*integer*	Indicates if Ipv6 multicast routing is enabled.

ipv6UcastRouting	*integer*	Indicates if Ipv6 unicast routing is enabled.
mtuId	*integer*	Hardware index corresponding to the MTU of the interface.
vrfId	*integer*	VRF id for this interface.

2.1288 show platform barefoot l3 mtu

[cmdErr]	*string*	Error Message.
[l3MtuInfo]	*list of MtuEntry* see 2.1288.1	List of mapping of MTU value to hardware MTU index.

2.1288.1 MtuEntry

mtuId	*integer*	Hardware MTU Index.
mtuValue	*integer*	MTU Value.

2.1289 show platform barefoot l3 next-hop

ucastNexthops	*table(numeric string) = UcastNexthop* see 2.1284.1.1	Complete list of mappings of id to unicast nexthop detail.

2.1290 show platform barefoot l3 next-hop NEXTHOPID

see 2.1284.1.1

2.1291 show platform barefoot l3 pipe allocation

[cmdErr]	*string*	Error Message.
pipes	*table(numeric string) = PipeResources* see 2.1291.1	Mapping of pipe ID to number of routes and number of VRFs.

2.1291.1 PipeResources

numRoutes	*integer*	Number of routes allocated.
numVrfs	*integer*	Number of VRFs allocated.

2.1292 show platform barefoot l3 route PREFIX [...]

[backoffInfo]	*BackoffInfo* see 2.1292.1	Back off programming info.
[cmdErr]	*string*	Error Message.
[routeConfigInfo]	*RouteConfigInfo* see 2.1292.2	Route programming info.

2.1292.1 BackoffInfo

nextHopId	*integer*	Nexthop index.
taskName	*string*	scheduler task backing off this route programming.

2.1292.2 RouteConfigInfo

[devId] **	*integer*	device ID. **Since: 4.27.1
inHostTable	*string*	(Y/N)route is in host table.
[localPartitionId] **	*integer*	per-instance partitionId. **Since: 4.25.2
nextHopId	*integer*	Nexthop index.
[partitionId]	*integer*	logical table partitionId.
[pipeId] **	*integer*	pipeline ID. **Since: 4.27.1
[prefix]	*string*	hw prefix representation.
[tableInstance] **	*integer*	partition table instance. **Since: 4.25.2
tableName	*string*	logical table where route is programmed.

2.1293 show platform barefoot l3 route last-hit (vrf (VRF | all)) [...]

[cmdErr]	*string*	Error Message.
[vrfs]	*table(string) = VrfRoutesLastHit* see 2.1293.1	Mapping of VRF name to IP routes.

2.1293.1 VrfRoutesLastHit

[routes]	*table(IPv4 or IPv6 address prefix) = PrefixLastHit* see 2.1293.1.1	Mapping of prefix to last hit.

2.1293.1.1 PrefixLastHit

lastHit	*string*	Last Hit.

2.1294 show platform barefoot l3 route last-hit START_TIME END_TIME [...]

see 2.1293

2.1295 show platform barefoot l3 route last-hit summary [...]

[cmdErr]	*string*	Error Message.
[lasthit]	*table(string) = NumRoutes* see 2.1295.1	Mapping of time slices to num routes.

2.1295.1 NumRoutes

numRoutes	*integer*	Number of routes in time slice.

2.1296 show platform barefoot l3 scheduler [...]

see 2.79

2.1297 show platform barefoot l3 scheduler backlog TASKNAME

[cmdErr]	*string*	Error Message.
[routeBacklog]	*list of RouteBacklogEntry* see 2.1297.1	List of routes waiting to be processed by this scheduler task.

2.1297.1 RouteBacklogEntry

nextHopId	*integer*	Nexthop index.
prefix	*IP address prefix*	IP prefix.
vrfId	*integer*	VRF id for this route.

2.1298 show platform barefoot l3 summary

ecmpGroupsTotal	*integer*	Total number of ECMP groups.
ecmpMembersTotal	*integer*	Total number of ECMP members.
ipv4RoutesInBackoff	*integer*	Total number of IPv4 routes waiting in the backoff queue, for other routes or adjacencies to be programmed.
ipv4RoutesTotal	*integer*	Total number of IPv4 routes.
ipv6RoutesInBackoff	*integer*	Total number of IPv6 routes waiting in the backoff queue, for other routes or adjacencies to be programmed.
ipv6RoutesTotal	*integer*	Total number of IPv6 routes.
l3InterfacesTotal	*integer*	Total number of L3 interfaces.
l3SubinterfacesTotal	*integer*	Total number of L3 Subinterfaces.
nextHopsTotal	*integer*	Total number of nexthops.
partitionUsage	*table(string) = PartitionUsage* see 2.1298.1	A mapping of lpm table to its partition usage.
reservedNextHopsTotal	*integer*	Total number of reserved nexthops.
routerNextHopsTotal	*integer*	Total number of router nexthops.
tableSummary	*table(string) = TableSummary* see 2.1282.1	A mapping of hardware table to its usage statistics.
tunnelNextHopsTotal	*integer*	Total number of tunnel nexthops.

2.1298.1 PartitionUsage

free	*integer*	Number of free partition ids.
used	*integer*	Number of partition ids in use.

2.1299 show platform barefoot l3 summary pipes [...]

[cmdErr]	*string*	Error Message.
tables	*table(string) = PipeUsageInfo* see 2.1299.1	A mapping of hardware table to its per pipe usage statistics.

2.1299.1 PipeUsageInfo

pipeUsage	*table(string) = TableSummary* see 2.1282.1	A mapping of a pipe to its usage statistics.

2.1300 show platform barefoot l3 task stats summary

taskSummary	*table(string) = TaskStatsSummaryEntry* see 2.1300.1	A mapping from task-name to its timing stats.

2.1300.1 TaskStatsSummaryEntry

taskStats	*table(string) = TaskStats* see 2.1300.1.1	A mapping from stat-type to timing stats.

2.1300.1.1 TaskStats

avgTime	*integer*	Average time in micro seconds.
count	*integer*	Task count.
maxTime	*integer*	Max time in micro seconds.
minTime	*integer*	Min time in micro seconds.

2.1301 show platform barefoot l3 vrf-mapping [...]

[cmdErr]	*string*	Error Message.
vrfs	*table(string) = VrfToPipe* see 2.1301.1	Mapping of VRF name to VRF ID and pipe ID.

2.1301.1 VrfToPipe

pipeId	*integer*	Pipe ID.
vrfId	*integer*	VRF ID.

2.1302 show platform barefoot lb errors

see 2.870

2.1303 show platform barefoot lb hash DIP SIP DPORT SPORT PROTO

hashValue	*integer*	Hash Value.

2.1304 show platform barefoot lb nexthop

[cmdErr]	*string*	Error message.
[servers]	*table(IPv4 or IPv6 address) = LbNexthopEntry* see 2.1304.1	Mapping of server IP address to next-hop index.

2.1304.1 LbNexthopEntry

nexthopId	*integer*	Next-hop index.
refCount	*integer*	Next-hop reference count.

2.1305 show platform barefoot lb nexthop-group NEXTHOP_GROUP hash HASH

[cmdErr]	*string*	Error message.
compressed	*boolean*	Entry exists in the compressed partition of the LbLookup block.
hwHash	*integer*	Final hash value used in table lookup.
lbGroupId	*integer*	Load Balancer group ID.
[maxVersion]	*integer*	Maximum connection-history version programmed in hardware.
[minActiveVersion]	*integer*	Minimum connection-history version that is not expiring in RTT.
[minVersion]	*integer*	Minimum connection-history version programmed in hardware.
nexthopId	*integer*	Nexthop ID in hardware corresponding to the server.
serverAddress	*IP address*	Server address.
tableId	*integer*	LbLookup table instance.
tableOffset	*integer*	Entry ID in the LbLookup table.
[versionInfo]	*table(numeric string) = LbConnectionHistoryVersionInfo* see 2.1305.1	Mapping of a connection-history version to the corresponding server details.

2.1305.1 LbConnectionHistoryVersionInfo

entryType	*enum: rtt | inHardware | pending*	Entry type.
expirationTime	*floating point number*	Expiration time in seconds for this connection-history entry.
nexthopId	*integer*	Nexthop ID in hardware corresponding to the server.
serverAddress	*IP address*	Server address.

2.1306 show platform barefoot lb nexthop-group NEXTHOP_GROUP_NAME

[cmdErr]	*string*	Error message.
[l3Section]	*L3Section* see 2.1306.1	L3 information.
[l3SectionErr]	*string*	Error message related to the L3 section.
[lbSection]	*LbSection* see 2.1306.2	lb information.
[lbSectionErr]	*string*	Error message related to the LB section.

2.1306.1 L3Section

[configSystemGenTime]	*floating point number*	L3 configuration system gentime.
configVersion	*integer*	L3 configuration version.
[statusSystemGenTime]	*floating point number*	L3 status system gentime.
[statusVersion]	*integer*	L3 status version.
[viaInfo]	*table(IPv4 or IPv6 address) = ViaInfo* see 2.1306.1.1	Mapping between IP addresses and their via information.

2.1306.1.1 ViaInfo

adminState	*enum: shutdown | enabled*	Admin state.
entryIds	*list of NexthopGroupEntryId* see 2.1306.1.1.1	Nexthop group entry ID list.
nexthopId	*integer*	Nexthop ID.
operState	*enum: down | up*	Operational state.
weight	*integer*	Nexthop weight.

2.1306.1.1.1 NexthopGroupEntryId

entryId	*integer*	Nexthop group entry ID.

2.1306.2 LbSection

hwFecId	*integer*	Hardware FEC ID associated with the nexthop group.
lbGrpId	*integer*	Load Balancer group Id.
maxGroupSize	*integer*	Nexthop group max size.
startOffset	*integer*	Starting hardware lookup table offset.
tableId	*integer*	Hardware lookup table Id.

2.1307 show platform barefoot lb nexthop-group usage

[lbNhgUsage]	*table(numeric string) = LbNhgUsageEntry* see 2.1307.1	Mapping between group size and the usage.

2.1307.1 LbNhgUsageEntry

available	*integer*	Number of groups available.
used	*integer*	Number of groups used.

2.1308 show platform barefoot lb summary

connHistoryTablesTotal	*integer*	Total number of connection history entries.
lookupTablesTotal	*integer*	Total number of lookups across all lookup tables.
tableSummary	*table(string) = TableSummary* see 2.1282.1	A mapping of hardware table to its usage statistics.

2.1309 show platform barefoot load-balance profile

chips	*table(string) = BfnHashProfile* see 2.1309.1	Collection of load-balance hashing profiles configured, indexed by the chip names.

2.1309.1 BfnHashProfile

hashSeed	*integer*	Hash seed.
ipv4	*LoadBalanceFieldsIpv4Header* see 2.1309.1.1	IPv4 hash fields settings.
ipv6	*LoadBalanceFieldsIpv6Header* see 2.1309.1.2	IPv6 hash fields settings.
l4	*LoadBalanceFieldsL4Header* see 2.1309.1.3	L4 hash fields settings.
lbProfile	*string*	Load balance profile configured.
mac	*LoadBalanceFieldsMacHeader* see 2.1309.1.4	MAC hash fields settings.

2.1309.1.1 LoadBalanceFieldsIpv4Header

ipDst	*boolean*	Hashing on destination IPv4 address.
ipSrc	*boolean*	Hashing on source IPv4 address.
protocol	*boolean*	Hashing on protocol.

2.1309.1.2 LoadBalanceFieldsIpv6Header

flowLabel	*boolean*	Hashing on flow label.
ipDst	*boolean*	Hashing on destination IPv6 address.
ipSrc	*boolean*	Hashing on source IPv6 address.
nextHeader	*boolean*	Hashing on next header.

2.1309.1.3 LoadBalanceFieldsL4Header

portDst	*boolean*	Hashing on destination port.
portSrc	*boolean*	Hashing on source port.

2.1309.1.4 LoadBalanceFieldsMacHeader

ethType	*boolean*	Hashing on ether type.
ingressPort	*boolean*	Hashing on ingress port.
macDst	*boolean*	Hashing on destination MAC address.
macSrc	*boolean*	Hashing on source MAC address.

2.1310 show platform barefoot mirroring session

revision 2

eosSessions	*table(string) = integer*	EOS session values.
sessions	*table(numeric string) = BfnMirrorSession* see 2.1310.1	Bfn Mirror sessions.

2.1310.1 BfnMirrorSession

direction	*enum: directionTx | directionRx | directionBoth*	Direction of mirroring.
mirrorDestSelector	*table(numeric string) = BfnMirrorDestSelector* see 2.1310.1.1	Map of mcGroupId to BfnMirrorDestSelector.
mirrorSourceList	*list of BfnMirrorSource* see 2.1310.1.2	List of Mirror sources.
sessionId	*integer*	Bfn Session Id.

2.1310.1.1 BfnMirrorDestSelector

mcGroupId	*integer*	Multicast group Id.
mirrorBufferId	*integer*	Mirror buffer Id.
mirrorDest	*table(numeric string) = BfnMirrorDest* see 2.1310.1.1.1	Map of McRid to BfnMirrorDest.
sflowMirrorBufferId **	*integer*	Sflow Mirror buffer Id. **Since: 4.24.0

2.1310.1.1.1 BfnMirrorDest

encapType	*enum: encapTypeCpu | encapTypeInt | encapTypeNone | encapTypeGre*	Mirror packet encapsulation.
mcRid	*integer*	Multicast Rid.
mirrorDestId	*BfnMirrorDestId* see 2.1310.1.1.1.1	Mirror destination.
portIndex	*integer*	Hardware Port index.

2.1310.1.1.1.1 BfnMirrorDestId

eosSessionId	*integer*	Eos Session Id.
intfId	*interface name*	Destination Interface Id.

2.1310.1.2 BfnMirrorSource

aclName	*string*	Acl name.
direction	*enum: directionTx | directionRx | directionBoth*	Direction of mirroring.

intfId	*interface name*	Source Interface Id.

2.1311 show platform barefoot port info

interfaces	*table(Interface name) = PortInfoEntry* see 2.1311.1	Dictionary mapping an interface to its hardware properties.

2.1311.1 PortInfoEntry

deviceId	*integer*	Device ID.
ingVPort	*integer*	Ingress virtual port.
pipeId	*integer*	Pipe ID.
pipePortIndex	*integer*	Port index local to the given pipe.
portIndex	*integer*	Port index.

2.1312 show platform barefoot profile

profiles	*table(string) = BfnProfileChipsPipeRoleModel* see 2.1312.1	Mapping of profile names to which chips use those profiles.

2.1312.1 BfnProfileChipsPipeRoleModel

chips	*list of string*	Chip this profile applies to.
pipes	*list of string*	List of pipes that the specified profile runs on.
role	*enum: Custom | Partner | Hybrid | EOS*	Whether its an EOS or custom program.

2.1313 show platform barefoot segment-security counters drop

vrfs	*table(string) = VrfHardwareDropCounters* see 2.1313.1	A mapping of VRF name to its drop counters.

2.1313.1 VrfHardwareDropCounters

sourceSegments	*table(string) = SourceSegmentDropCounters* see 2.1313.1.1	A mapping of source segment to drop counters.

2.1313.1.1 SourceSegmentDropCounters

policyBytes	*integer*	Policy-dropped byte counters.
policyPkts	*integer*	Policy-dropped packet counters.
rpfBytes	*integer*	RPF-dropped byte counters.
rpfPkts	*integer*	RPF-dropped packet counters.

2.1314 show platform barefoot sflow summary

egressIntfs	*table(string) = SflowConfigAndHwState* see 2.1314.1	Egress sflow interfaces.
ingressIntfs	*table(string) = SflowConfigAndHwState* see 2.1314.1	Ingress sflow interfaces.

2.1314.1 SflowConfigAndHwState

configState	*enum: disabled | enabled*	Sflow enabled or disabled for interface in config.
hwState	*enum: sampling | notSampling*	Sflow sampling or not for interface in hardware.
[vlan]	*integer*	Internal VLAN for this interface.

2.1315 show platform barefoot system

[chips]	*table(numeric string) = BfnSystemEntry* see 2.1315.1	Dictionary mapping a device ID to its chip hardware properties.
[errorMessage]	*string*	Error message if not successful.

2.1315.1 BfnSystemEntry

chipName	*string*	Chip name.
chipPart	*string*	Chip part.
chipRevStr	*string*	Chip revision.
chipRole	*string*	Chip role.
chipVariant **	*string*	Chip variant. **Since: 4.21.3
deviceId	*integer*	Device ID.
profileName	*string*	Profile name.
sliceName	*string*	Slice name.

2.1316 show platform barefoot traffic-manager queue status [...]

interfaces	*table(Interface name) = QueuesInIntf* see 2.1316.1	Dictionary mapping an interface to the corresponding Queues.

2.1316.1 QueuesInIntf

queues	*table(string) = QueueData* see 2.1316.1.1	Dictionary mapping a Queue to the corresponding Queue status data.

2.1316.1.1 QueueData

bytesUsed	*integer*	Number of bytes used.
dynamicThreshold	*floating point number*	Dynamic threshold percentage.
minReservedBytes	*integer*	Minimum reserved bytes.

2.1317 show platform barefoot vlan membership DIRECTION

see 2.1281

2.1318 show platform barefoot vlan table l2DomainLookup

devices	*table(numeric string) = DeviceL2DomainLookupTable* see 2.1318.1	Dictionary mapping a device ID to the corresponding l2DomainLookup table instances.

2.1318.1 DeviceL2DomainLookupTable

pipes	*table(numeric string) = PipeL2DomainLookupTable* see 2.1318.1.1	Dictionary mapping a pipe ID to the corresponding l2DomainLookup table instance.

2.1318.1.1 PipeL2DomainLookupTable

entries	*table(numeric string) = L2DomainLookupEntry* see 2.1318.1.1.1	Dictionary mapping an l2DomainLookup table entry ID to its contents in a given pipe.

2.1318.1.1.1 L2DomainLookupEntry

action	*enum: vlanXlate | untagged | tagged*	Action taken if the incoming packet matches this entry.
ingVirtualPort	*integer*	Ingress virtual port of the packet.
l2Aintf	*integer*	L2 interface index.
priority	*integer*	Priority associated with the entry.
switchPort **	*boolean*	Ingress switch port of the packet. **Since: 4.22.0
vlanHdrValid	*boolean*	VLAN header of the matching packet.
vlanId	*integer*	VLAN identifier.
vlanIdMask	*integer*	Mask for the VLAN identifier.

2.1319 show platform barefoot vlan table vlanEncap

devices	*table(numeric string) = DeviceVlanEncapTable* see 2.1319.1	Dictionary mapping a device ID to the corresponding vlanEncap table instances.

2.1319.1 DeviceVlanEncapTable

pipes	*table(numeric string) = PipeVlanEncapTable* see 2.1319.1.1	Dictionary mapping a pipe ID to the corresponding vlanEncap table instance.

2.1319.1.1 PipeVlanEncapTable

entries	*table(numeric string) = VlanEncapEntry* see 2.1319.1.1.1	Dictionary mapping a vlanEncap table entry ID to its contents in a given pipe.

2.1319.1.1.1 VlanEncapEntry

action	*enum: NoVlanEncap | NeedVlanEncap*	Action taken if the incoming packet matches this entry.
egressPort	*integer*	Egress interface.
vlanId	*integer*	VLAN identifier.

2.1320 show platform barefoot vlan table vlanProperties

propertyTables	*table(numeric string) = VlanPropertiesEntry* see 2.1320.1	Dictionary mapping a vlanProperties table entry ID to its contents.

2.1320.1 VlanPropertiesEntry

action	*string*	Action taken if the incoming packet matches this entry.
ingL2Domain	*integer*	Ingress L2 Domain.
snoopingEnabled	*boolean*	Flag which indicates whether IGMP snooping is enabled on the given VLAN.
sviEnabled	*boolean*	Flag which indicates whether a SVI is configured on the given VLAN.
vxlan **	*boolean*	Flag which indicates whether a vxlan is enabled for the given VLAN. **Since: 4.21.0

2.1321 show platform barefoot vxlan table intfOverlaySrcMacRewrite [...]

vtis	*table(string) = VtiVniToMacAndHwState* see 2.1321.1	Mapping of VTI to its VNIs.

2.1321.1 VtiVniToMacAndHwState

vnis	*table(numeric string) = RewriteSmacAndHwState* see 2.1321.1.1	Mapping of a VNI in a VTI to MAC address.

2.1321.1.1 RewriteSmacAndHwState

hwState	*enum: notProgrammed | programmed | misprogrammed*	Rewritten source MAC state in hardware table.
smac	*MAC address*	Rewritten source MAC address.

2.1322 show platform barefoot vxlan table overlaySrcMacRewrite [...]

vrfs	*table(string) = VrfIpToMac* see 2.1322.1	Mapping of VRF name to its overlay source IP addresses.

2.1322.1 VrfIpToMac

addresses	*table(IPv4 address) = RewriteSmac* see 2.1322.1.1	Mapping of IP to MAC address.

2.1322.1.1 RewriteSmac

smac	*MAC address*	Rewritten source MAC address.

2.1323 show platform bios

revision 2

[fallbackVersion]	*BiosVersion* see 2.1323.1	BIOS fallback version programmed on the switch.
[programmedVersion]	*BiosVersion* see 2.1323.1	BIOS version programmed on the switch.
runningVersion	*BiosVersion* see 2.1323.1	BIOS version currently running on the switch.
[standbyVersions]	*BiosVersions* see 2.1323.2	Versions on standby supervisor.
sysName	*string*	Name of the system.

2.1323.1 BiosVersion

changeNumber **	*string*	Change number. **Since: 4.26.2
[core]	*string*	Core string.
line **	*string*	Aboot line (eg. norcal9). **Since: 4.26.2
[pcieLaneConfig] **	*string*	PCIe lane configuration. **Since: 4.26.2
version **	*string*	Version string (eg. 7.2.1). **Since: 4.26.2
versionString **	*string*	Complete Aboot version string. **Since: 4.26.2

2.1323.2 BiosVersions

Could also be an instance of:
DualSupBiosVersions 2.1323

[fallbackVersion]	*BiosVersion* see 2.1323.1	BIOS fallback version programmed on the switch.
[programmedVersion]	*BiosVersion* see 2.1323.1	BIOS version programmed on the switch.
runningVersion	*BiosVersion* see 2.1323.1	BIOS version currently running on the switch.
sysName	*string*	Name of the system.

2.1324 show platform bios history [...]

revision 2

history	*list of HistoryEntry* see 2.1324.1.1	List of installation history.
[standbyHistory]	*BiosHistory* see 2.1324.1	History on standby supervisor.
sysName	*string*	Name of the system.

2.1324.1 BiosHistory

Could also be an instance of:
DualSupBiosHistory 2.1324

history	*list of HistoryEntry* see 2.1324.1.1	List of installation history.
sysName	*string*	Name of the system.

2.1324.1.1 HistoryEntry

checksum	*string*	Checksum of AUF used.
epoch	*integer*	Epoch of update.
error	*integer*	Error code.
name	*string*	Name of AUF used.
payloadType	*enum: aboot | other | certificateBundle*	Type of AUF used.

2.1325 show platform cortina INTF diag counters

[cmdErr]	*string*	The error message if any.
[interfaces]	*table(Interface name) = CountersModel* see 2.1325.1	A mapping of the interface to its counters.

2.1325.1 CountersModel

counters	*table(string) = integer*	Map of counter names and their values.

2.1326 show platform etba counters [...]

[ebraCounters]	*table(string) = HandlerCounter* see 2.1326.1	Maps EbraTestBridge routing handler names to respective counters.
ebraIgnored	*integer*	Number of packets that were ignored by all EbraTestBridge routing handlers.
ebraRouted	*integer*	Number of packets that were routed by at least one EbraTestBridge routing handler.
[fwdIntfCounters]	*table(string) = HandlerCounter* see 2.1326.1	Maps FwdIntfDevice routing handler names to respective counters.
fwdIntfIgnored	*integer*	Number of packets that were ignored by all FwdIntfDevice routing handlers.
fwdIntfRouted	*integer*	Number of packets that were routed by a FwdIntfDevice routing handler.

2.1326.1 HandlerCounter

ignored	*integer*	Number of ignored packets.
routed	*integer*	Number of routed packets.

2.1327 show platform fap ecmp hash [...]

faps	*table(string) = HashInfoFap* see 2.1327.1	Mapping of FAP names and cores to hash parameters and associated values.

2.1327.1 HashInfoFap

hashPolys	*list of integer*	Collection of hash polynomials that exist in the FAP.
hashSeeds	*list of integer*	Collection of hash seeds that exist in the FAP.
keyShifts	*list of integer*	Collection of hash key shifts that exist in the FAP.

2.1328 show platform fap port-channel hash [...]

see 2.1327

2.1329 show platform fap voq delete

voqDeleteEntries	*list of VoqDeleteEntry* see 2.1329.1	List of VOQ delete entries.

2.1329.1 VoqDeleteEntry

chipName	*string*	Ingress chip.
coreId	*integer*	Core ID.
count	*integer*	Count of VOQ deletes.
firstOccurrence	*floating point number*	UTC timestamp of the first delete occurrence for this VOQ.
intfName	*interface name*	Name of the interface.
lastOccurrence	*floating point number*	UTC timestamp of the last recorded VOQ delete.
trafficClass	*integer*	Traffic class.

2.1330 show platform ip igmp snooping groups [...]

vlans	*table(numeric string) = VlanStatus* see 2.1330.1	A mapping of VLAN ID to its Ethernet groups.

2.1330.1 VlanStatus

ethGroups	*table(MAC address) = VlanOrMcastGrpStatus* see 2.1330.1.1	Ethernet groups information, keyed by their address.

2.1330.1.1 VlanOrMcastGrpStatus

interfaces	*table(Interface name) = InterfaceModel* see 2.1330.1.1.1	A mapping of interface to its programmed status.
mcastId	*integer*	Multicast ID.
programmedIn	*enum: hardware | software | both*	Database in which the Ethernet group is programmed. 'software'- ethGroup programmed in software only. 'hardware'- ethGroup programmed in hardware only. 'both'- ethGroup present in both hardware & software.

2.1330.1.1.1 InterfaceModel

programmedIn	*enum: hardware | software | both*	Database in which the interface is programmed: 'software'- interface programmed in software only. 'hardware'- interface programmed in hardware only. 'both'- interface in both hardware & software.

2.1331 show platform module [...]

modules	*table(string) = PlatformModuleStatus* see 2.1331.1	Mapping between module name and module status.

2.1331.1 PlatformModuleStatus

status	*enum: empty | diags | unknown | inserted*	Status of module.

2.1332 show platform pkt

2.1333 show platform power controller

powerControllers	*list of PowerController* see 2.1333.1	List of power controller info.

2.1333.1 PowerController

modelName	*string*	power controller model name.
name	*string*	power controller name.
resetCount	*integer*	power controller reset count.
version	*string*	power controller version.

2.1334 show platform power isl

islControllers	*table(string) = IslController* see 2.1334.1	ISL controllers, keyed by their names.

2.1334.1 IslController

agentName	*string*	Name of ISL's agent.
controllerState	*string*	Controller state.
outputVoltageRails	*list of OutputVoltageRail* see 2.1334.1.1	Metrics for each ISL output.
vin	*floating point number*	Input voltage.

2.1334.1.1 OutputVoltageRail

[description]	*string*	Output description.
iout	*floating point number*	Measured output current.
online **	*boolean*	Device is online. **Since: 4.29.1
pout	*floating point number*	Measured output power.
vout	*floating point number*	Measured output voltage.
voutConfigured	*floating point number*	Configured output voltage.

2.1335 show platform ptp hardware-sync

configuredState	*enum: off | on*	Configured State.
masters	*table(string) = MasterModel* see 2.1335.1	A Mapping between a TimeSync Master and its time synchronization information.
operationalState	*enum: stable | disabled | enabled | active | synchronizing*	Operational State.
slaves	*table(string) = SlaveModel* see 2.1242.1	A Mapping between a TimeSync Slave and its time synchronization information.
syncInterval	*floating point number*	Synchronization Interval (in milliseconds).

2.1335.1 MasterModel

[localTod]	*integer*	Latest Local Time of Day (ToD).
[masterTod]	*integer*	Latest Master Time of Day (ToD).
[phaseDifferences]	*list of integer*	Last 4 phase differences (in ns).
pidState **	*enum: runServo | servoDisabled | watchSkew | adjustSkew | jumpClock*	Servo State. **Since: 4.20.5
[skew] **	*floating point number*	Active Master Skew. **Since: 4.20.5

state	*enum: stableStandby | switchover | initializedStandby | disabled | active | initialized | synchronizingStandby | failed*	Operational State.
[timeToStability]	*integer*	Time to stability (in s).
[updates]	*integer*	Sync Update(s).

2.1336 show platform ptp hardware-sync master SCD_NAME

masters	*table(string) = MasterModel* see 2.1335.1	A Mapping between a TimeSync Master and its time synchronization information.

2.1337 show platform ptp hardware-sync slave SWITCH_NAME

see 2.1242

2.1338 show platform sand capabilities

revision 2

aclHwStatusReporting-Supported	*boolean*	ACL hardware status reporting supported.
arpTableSize	*integer*	ARP table size.
asyncSupportRequired	*boolean*	QoS ACL asynchronous support required.
defaultCos	*integer*	Default CoS.
defaultCosSupported	*boolean*	Default CoS supported.
defaultDscp	*integer*	Default DSCP.
defaultDscpSupported	*boolean*	Default DSCP supported.
dot1qTunnelSupported	*boolean*	Dot1q tunnel supported.
dropsLogDirectionSupport	*enum: directionNone* | *directionTxOnly* | *directionTxAndRx*	Congestion drops log direction support.
dynamicExpMappingSupported	*boolean*	Dynamic MPLS EXP bits to/from TC mapping supported.
ecmpSupported	*boolean*	IPv6 ECMP supported.
ecnSupported	*boolean*	ECN supported.
egressArpTableSize	*integer*	Maximum egress ARP table entries.
fecForEachHop	*boolean*	FEC for each hop.
hwStatusReportingSupported	*boolean*	Hardware status reporting supported.
ingressPolicingSupported	*boolean*	Ingress policing supported.
ingressTrTcmPolicing-Supported	*boolean*	Ingress trTcm (two rate three color) policing supported.
ip6AclsSupported	*boolean*	IPv6 ACLs supported.
ipAclsSupported	*boolean*	IPv4 ACLs supported.
ipv6AclPbrSupported	*boolean*	IPv6 ACL PBR supported.
ipv6NexthopPbrSupported	*boolean*	IPv6 nexthop PBR supported.
lagMulticastLoadBalancing	*boolean*	LAG multicast load balancing.
maxEcmp	*integer*	Maximum ECMP.
maxIpsecTunnelInterfaces	*integer*	Maximum IP Security tunnel interfaces.
maxLags	*integer*	Maximum LAGs.
maxMacTableSize	*integer*	Maximum MAC table entries.
maxMembersPerLag	*integer*	Maximum members per LAG.
maxMulticastGroups	*integer*	Maximum multicast groups.
maxNexthopGroupEcmp	*integer*	Maximum nexthop group ECMP.
maxShapeRatePps	*integer*	Maximum shape rate (pps).
maxSvi	*integer*	Maximum SVI.
maxSwLags	*integer*	Maximum software LAGs.
maximumTopologiesSupported	*integer*	Maximum topologies supported.
minPortShapeRate	*integer*	Minimum port shape rate (kbps).
minTxQueueShapeRate	*integer*	Minimum transmit queue shape rate (kbps).
mplsMatchRulePbrSupported	*boolean*	MPLS match rule PBR supported.
mplsPseudowireSupported	*boolean*	MPLS pseudowire supported.
mplsStaticPseudowire-Supported **	*boolean*	MPLS static pseudowire supported. **Since: 4.30.1
msrpSupported	*boolean*	MSRP supported.
mvrpSupported	*boolean*	MVRP supported.

neighborTableSize	*integer*	ARP neighbor table size.
nexthopGroupMplsLabelStack-Size	*integer*	Nexthop group MPLS label stack entries.
nexthopGroupPbrSupported	*boolean*	Nexthop group PBR supported.
nexthopGroupSupported	*boolean*	Nexthop group supported.
nexthopGroupTypeSupported	*table(string) = boolean*	A mapping between nexthop group type and if it is supported.
pbrNexthopVrfSupported	*boolean*	PBR Nexthop Vrf supported.
pbrSupported	*boolean*	PBR supported.
policeByteModeMaxBurstBytes	*integer*	Police byte mode maximum burst bytes.
policeByteModeMaxRateKbps	*integer*	Police byte mode maximum rate (kbps).
policeByteModeMinBurstBytes	*integer*	Police byte mode minimum burst bytes.
policeByteModeMinRateKbps	*integer*	Police byte mode minimum rate (kbps).
policyQosActionsSupported	*table(string) = boolean*	A mapping between Policy QoS actions and if it is supported.
policyQosNumCMapSupported	*integer*	Number of QoS policy class maps supported.
policyQosSupported	*boolean*	Policy QoS supported.
routingMplsVpnV4	*boolean*	MPLS VPNv4 supported.
routingMplsVpnV6	*boolean*	MPLS VPNv6 supported.
sandDanzRunnable	*boolean*	Sand DANZ runnable.
sandIpsecRunnable	*boolean*	Sand Ipsec runnable.
sflowDscpRewriteSupported	*boolean*	Sflow DSCP rewrite supported.
sharedPolicerSupported	*boolean*	Shared policer supported.
tcToDscpMapSupported	*boolean*	TC to DSCP map supported.
tunnelInterfaceModes-Supported	*table(string) = boolean*	Supported tunnel interface modes.
txQueueShapeRateUnitPps-Supported	*boolean*	Transmit queue shape rate unit (pps) supported.
urpfSupportedOnAllFaps	*boolean*	URPF supported on all FAPs.
usableLags	*integer*	Usable LAGs.
v6maxEcmp	*integer*	IPv6 Maximum ECMP.
vrfRoutingSupported	*boolean*	VRF routing supported.
vtepGroupSupported	*boolean*	VXLAN L2 ECMP supported.
vxlanBfdSupported	*boolean*	VXLAN bfd supported.
vxlanMcastGroupSupported	*boolean*	VXLAN multicast group supported.
vxlanSupported	*boolean*	VXLAN supported.
wredSupported	*boolean*	WRED supported.

2.1339 show platform sand fabric monitor capacity

fabricCapacityMonitorLog	*boolean*	Fap fabric capacity monitor log enable config.
fabricCapacityMonitor-Threshold	*integer*	Fap fabric capacity monitor threshold.
fapFabricCapacityInfo	*table(string) = FapFabricCapacityInfo* see 2.1339.1	Maps FapName to Fap fabric capacity info.

2.1339.1 FapFabricCapacityInfo

lastFabricCapacityMonitor-Event	*string*	Fabric Capacity monitor event.
lastFabricCapacityMonitor-EventTime	*string*	Last time fabric Capacity %% crossed above/below configiruredthreshold.
lastFullFabricCapacity **	*integer*	Full Fabric capacity. **Since: 4.25.0
lastFullFapCapacity	*integer*	Full Fap capacity.
lastOperationalFabric-Capacity	*integer*	Operational Serdess capacity.
lastOperationalFapCapacity	*integer*	Operational Fap capacity.

2.1340 show platform sand health

revision 2

dataGenerationTime	*floating point number*	Time at which data was generated in UTC.
fabricAuditLog **	*list of EventLogModel* see 2.1340.1	Fabric audit events. **Since: 4.30.1
fabricCardsNotInitialized	*table(string) = string*	A mapping of fabric card name to its hardware state.
fabricInterruptOccurrences	*table(string) = FabricInterruptDataModel* see 2.1340.2	A mapping of fabric name to its last interrupt data.
fabricMeshLog **	*list of EventLogModel* see 2.1340.1	Fabric mesh events. **Since: 4.30.1
fabricSerdesIsolated	*table(string) = PeerLinkId* see 2.1340.4	A mapping of chip name to isolated fabric SerDes.
linecardsNotInitialized	*table(string) = string*	A mapping of linecard name to its hardware state.
numFabricCards	*integer*	Number of fabric cards.
numLinecards	*integer*	Number of linecards.
shutdownSerdes	*table(string) = PeerLinkId* see 2.1340.4	A mapping of chip name to SerDes.
suspiciousRxFaps	*table(string) = FapDataModel* see 2.1340.3	A mapping of FAP name to number of times it's caused RQP discards.
suspiciousTxFaps	*table(string) = FapDataModel* see 2.1340.3	A mapping of FAP name to number of times it's caused RQP discards.
totalSerdesAdminShutdowns	*integer*	Fabric SerDes admin shutdown.

2.1340.1 EventLogModel

eventMessage	*string*	Event log.
eventTime	*floating point number*	Event time.

2.1340.2 FabricInterruptDataModel

count	*integer*	Number of occurrences.
[lastOccurrence]	*floating point number*	Time since fabric interrupt occurrence.

2.1340.3 FapDataModel

count	*integer*	Number of occurrences.

2.1340.4 PeerLinkId

peerLinks	*table(numeric string) = PeerLinkIdSubModel* see 2.1340.4.1	A mapping of link ID to peer Link.

2.1340.4.1 PeerLinkIdSubModel

[count]	*integer*	Number of times serDes link has been isolated.
[lastOccurrence]	*floating point number*	Last Occurrence.
name	*string*	Peer Link.

2.1341 show platform sand lag hardware-only

isHwOnly	*boolean*	Indicates all LAGs use hardware resources.

2.1342 show platform sand mcast replication ingress maximum

config	*integer*	Configured maximum number of ingress replications.
status	*integer*	Actual maximum number of ingress replications.

2.1343 show platform sand mdb profile [...]

revision 2

fapProfiles	*table(string) = MdbFapProfile* see 2.1343.1	MDB profile for each FAP, keyed by FAP name.
[systemProfile]	*enum: l3-xl | l3-xxl | balanced | l3 | default | balanced-xl | l3-xxxl*	MDB system profile.

2.1343.1 MdbFapProfile

fapId	*integer*	FAP ID.
fapType	*string*	FAP type.
profile	*string*	MDB Profile.

2.1344 show platform sand scheduler interface INTF

clSchedulers	*table(string) = ClScheduler* see 2.1344.1	CL (Class) scheduler per scheduler index.
coreId	*integer*	Local core ID for the given interface.
cores	*table(string) = VoqIngressCore* see 2.1344.5	VOQ flows per ingress core.
fapId	*integer*	FAP ID for the given interface.
fqSchedulers	*table(string) = FqScheduler* see 2.1344.2	FQ (Fair Queue) scheduler per scheduler index.
hrScheduler	*HrScheduler* see 2.1344.3	HR (Heirarchial) scheduler.
parentInterface	*InterfaceSubmodel* see 2.1344.4	Parent interface.
[subInterface]	*InterfaceSubmodel* see 2.1344.4	Subinterface.

2.1344.1 ClScheduler

classSchedMode	*enum: (SP,WFQ,WFQ,WFQ) | WFQ | (SP,SP,WFQ,WFQ) | SP | Unsupported*	Class scheduler mode.
cos	*integer*	CoS value (weight).
creditRate	*integer*	Credit rate in bps.
flowControlCounter	*integer*	Counts the number of clocks that flow control was issued at the port.
flowId	*integer*	Flow ID.
flowState	*enum: slow | off | on*	Flow state.
parentSE	*integer*	Parent scheduler element.
schedId **	*integer*	scheduler ID. **Since: 4.30.0
wfqMode	*enum: independent | discrete | perClass*	WFQ mode.

2.1344.2 FqScheduler

compositeFlow	*boolean*	Current flow is a composite flow.
cos	*integer*	CoS value (weight).
creditRate	*integer*	Credit rate in bps.
flowId	*integer*	Flow ID.
flowState	*enum: slow | off | on*	Flow state.
parentSE	*integer*	Parent scheduler element.

2.1344.3 HrScheduler

cos	*integer*	CoS value (weight).
creditRate	*integer*	Credit rate in bps.
flowId	*integer*	Flow ID.

schedulerIndex	*integer*	HR scheduler index.

2.1344.4 InterfaceSubmodel

name	*interface name*	Interface name.
sppid	*integer*	Interface system physical port ID.

2.1344.5 VoqIngressCore

flows	*table(string) = VoqFlow* see 2.1344.5.1	VOQ flows for this ingress core.

2.1344.5.1 VoqFlow

cos	*integer*	CoS value (weight).
creditRate	*integer*	Credit rate in bps.
voqId	*integer*	VOQ Id corresponding to the intf and core.

2.1345 show platform sand subinterface anchor [...]

interfaces	*table(Interface name) = Interface name*	A mapping of subinterface to its anchor interface.

2.1346 show platform sand subinterface capabilities

maxPortShapeRate	*integer*	Maximum SubInterface port shape rate (kbps).
maxSubIntfsWithVoqs	*integer*	Maximum number of subInterfaces with VOQ supported.
maxSubIntfsWithVoqsPerCore	*table(numeric string) = integer*	A mapping between system core and max subInterfaces with voq supported.
minPortShapeRate	*integer*	Minimum SubInterface port shape rate (kbps).
numVoqsPerSubIntf	*integer*	Number of Voqs per SubInterface.
schedulingMode	*table(string) = string*	A mapping between traffic class and scheduling mode.
tcToVoqMapping	*table(string) = string*	A mapping between traffic class and Voq.
voqTailDropThresholdBD	*integer*	Voq tail drop threshold in Buffer descriptors.
voqTailDropThresholdMB	*floating point number*	Voq tail drop threshold in MBs.

2.1347 show platform sand tapagg [...]

revision 5

groups	*table(numeric string) = TapAggGroup* see 2.1347.1	Tap-agg groups keyed by groupId.
nexthopGroups	*table(string) = TapAggNexthopGroup* see 2.1347.2	Tap-agg nexthop groups keyed by name.
tapPorts	*table(Interface name) = TapPort* see 2.1347.3	Dictionary of tap port configurations keyed by interface name.
toolPorts	*table(Interface name) = ToolPort* see 2.1347.4	Dictionary of tool ports configurations keyed by interface name.

2.1347.1 TapAggGroup

destinationId	*integer*	Destination ID.
destinationType	*enum: drop | port | fec | mcg | lag*	Destination Type.
[egressTruncation-DestinationId]	*integer*	Destination ID for egress truncation.
[egressTruncationIngress-Rep]	*boolean*	Ingress replication enabled for egress truncation destinations.
[egressTruncationToolPorts]	*list of Interface name*	List of egress truncated tool ports.
[groupName]	*string*	Group name.
ingressRep	*boolean*	Ingress replication enabled.
[notEgressTruncation-DestinationId]	*integer*	Destination ID for egress non-egress truncated ports.
[notEgressTruncation-IngressRep]	*boolean*	Ingress replication enabled for non-egress truncated destinations.
[notEgressTruncationTool-Ports]	*list of Interface name*	List of non-egress truncated tool ports.
[outlif]	*integer*	OutLIF allocated for tool group.
sourceInterfaces	*list of Interface name*	List of source interfaces.
[sourcePolicies]	*list of string*	List of source policies.
toolPorts	*list of Interface name*	List of tool ports.

2.1347.2 TapAggNexthopGroup

destinationId	*integer*	Destination ID.
destinationType	*enum: drop | port | fec | mcg | lag*	Destination Type.
egressInterfaces	*list of NhgEgressInterface* see 2.1347.2.1	List ofnexthop group egress interfaces.
nexthopGroupNames	*list of string*	List of nexthop group names.
sourceInterfaces	*list of Interface name*	List of source interfaces.

2.1347.2.1 NhgEgressInterface

intf	*interface name*	Interface for traffic destination.
outLif	*integer*	Logical output interface for traffic destination.

2.1347.3 TapPort

defaultGroup	*string*	The default group associated with a tap port.
greStrip	*boolean*	Remove GRE encapsulation header.
innerPortIdentity	*integer*	The inner port identity of a tap port.
[lag]	*interface name*	Port-channel this interface belongs to.
mplsPop	*boolean*	Remove MPLS labels.
portIdentity	*integer*	The port identity of a tap port.
truncationSize	*integer*	The size to which packet is to be truncated, in bytes.
vxlanStrip	*boolean*	Remove VXLAN header.

2.1347.4 ToolPort

brStrip	*boolean*	Remove BR Tag.
dot1qStrip	*list of integer*	The set of 802.1Q tags to remove on a tool port.
identityMode	*enum: dot1q | qinq | none*	Identity tagging mode.
[lag]	*interface name*	Port-channel this interface belongs to.
mplsPop	*boolean*	Remove MPLS labels.
[outLif]	*integer*	Logical output interface for traffic destination.
truncationSize	*integer*	The size to which packet is to be truncated, in bytes.
vnStrip	*boolean*	Remove VN Tag.

2.1348 show platform sand tapagg truncation-profile

revision 2

truncationProfileFapModel	*table(numeric string) = TruncationProfileFapModel* see 2.1348.1	Dictionary of truncation profile fap models keyed by fapId.

2.1348.1 TruncationProfileFapModel

fapName	*string*	Name of the fap.
numFree	*integer*	Number of free profiles.
truncationProfileInfoModel	*table(numeric string) = TruncationProfileInfoModel* see 2.1348.1.1	Dictionary of truncation profile info model keyed by profileId.

2.1348.1.1 TruncationProfileInfoModel

destination	*integer*	Destination ID.
groups	*table(numeric string) = string*	Group names keyed by groupId.
profileId	*integer*	Profile ID.
tapPortList	*list of string*	List of tap ports.
toolPortList	*list of Interface name*	List of tool ports.

2.1349 show platform sand traffic-class to network-qos map

tcToNwkQos	*table(numeric string) = integer*	Mapping between Traffic-class to network-qos.

2.1350 show platform sand voq tail-drop

ingressThresholds	*table(string) = IngressThresholds* see 2.1350.1	Arad ingress thresholds for each speed. Also includes ingress thresholds for speed and traffic class if any configured.

2.1350.1 IngressThresholds

fadtAlpha **	*integer*	Fair adaptive drop threshold scaling factor (alpha) for dynamic packet queue size. **Since: 4.20.0
fadtMinQSize **	*integer*	Minimum packet queue size for fair adaptive drop threshold. **Since: 4.20.0
[guaranteedBuff]	*integer*	Number of guaranteed buffers for Packet Queue.
pqMaxQBuffSize	*integer*	Packet Queue Buffer size thresholds for DP Normal.
pqMaxQBuffSizeCpu	*integer*	Packet Queue Buffer size thresholds for DP Cpu.
pqMaxQSize	*integer*	Packet Queue size thresholds for DP Normal.
pqMaxQSizeCpu	*integer*	Packet Queue size thresholds for DP Cpu.
[trafficClassThresholds]	*table(numeric string) = IngressTcThresholds* see 2.1350.1.1	Arad ingress thresholds for speed and traffic class. Key is traffic class number 0-7.

2.1350.1.1 IngressTcThresholds

fadtAlpha **	*integer*	Fair adaptive drop threshold scaling factor (alpha) for dynamic packet queue size. **Since: 4.20.0
fadtMinQSize **	*integer*	Minimum packet queue size for fair adaptive drop threshold. **Since: 4.20.0
[guaranteedBuff]	*integer*	Number of guaranteed buffers for Packet Queue.
[guaranteedBuffActive]	*table(string) = IngressGuaranteedBuff* see 2.1350.1.1.1	Mapping of linecard to active guaranteed buffer value. Key ALL - all linecards, Key LinecardN - linecard <N> .
pqMaxQBuffSize	*integer*	Packet Queue Buffer size thresholds for DP Normal.
pqMaxQBuffSizeCpu	*integer*	Packet Queue Buffer size thresholds for DP Cpu.
pqMaxQSize	*integer*	Packet Queue size thresholds for DP Normal.
pqMaxQSizeCpu	*integer*	Packet Queue size thresholds for DP Cpu.

2.1350.1.1.1 IngressGuaranteedBuff

runningGuaranteedBuff	*integer*	Active guaranteed buffer value.

status	*enum: inactive | active*	inactive - reload linecard for guaranteed buffer configuration to take effect. active - linecard running with configured guaranteed buffer.

2.1351 show platform sand voq tail-drop fadt

ingressThresholds	*table(string) = IngressThresholds* see 2.1350.1	Arad ingress thresholds for each speed. Also includes ingress thresholds for speed and traffic class if any configured.

2.1352 show platform sand voq tail-drop fadt status

fadtStatus	*table(string) = string*	FADT enabled status for each FAP.

2.1353 show platform scd satellite debug [...]

scds	*table(string) = SatelliteScd* see 2.1353.1	Satellite Scds.

2.1353.1 SatelliteScd

collectionTime	*string*	Collection time.
name	*string*	Satellite Scd Name.
registers	*table(numeric string) = DebugRegister* see 2.1353.1.1	Debug registers.

2.1353.1.1 DebugRegister

name	*string*	Register name.
offset	*integer*	Register offset.
value	*integer*	Register value.

2.1354 show platform sfe counters [...]

counters	*table(numeric string) = Counter* see 2.1354.1	A list of counters used by Bessd keyed by the counterId.
owners	*table(numeric string) = string*	A list of counter owners keyed by the ownerId.

2.1354.1 Counter

count	*integer*	Number of occurrences.
counterType	*enum: module | gatehook*	Counter type.
[name]	*string*	Name of the counter.

[ownerId]	*integer*	ID used to look up the module owner of the counter in the owners dictionary.
unit	*enum: memoryBlocks* | *bytes* | *packets*	Basic unit of the counter.

2.1355 show platform sfe flow tracking counters

exportInfoInRspAdd	*integer*	Number of add export message received.
exportInfoInRspRemove	*integer*	Number of remove export message received.
exportInfoInRspRemoveNoFlow	*integer*	Number of remove export message received for invalid flow.
exportInfoReq	*integer*	Number of export GRPC request message sent.
exportInfoRsp	*integer*	Number of export GRPC response message received.
exportInfoRspInvalidFlow **	*integer*	Number of export received for stale flows. **Since: 4.26.2
exportInfoRspInvalidFt	*integer*	Number of export received for invalid flow traker.
exportInfoRspInvalidIntf	*integer*	Number of export received for invalid flow tracker interface.
exportInfoRspNoModule	*integer*	Number of export received for invalid flow tracker module.
exportQueuePushAttempt **	*integer*	Number of export record enqueue attempted. **Since: 4.24.2
exportQueuePushFail	*integer*	Number of export record enqueue failed.
exportQueueWriteUnavailable **	*integer*	Number of times export queue write unavailable. **Since: 4.24.2
exportRingAddEnqFail	*integer*	Number of BESS export ring add enqueue failure.
exportRingDelEnqFail	*integer*	Number of BESS export ring delete enqueue failure.
exportRingInuse	*integer*	Number of export ring data in use.
exportRingSize	*integer*	Size of BESS export ring.
exportfd	*integer*	Export ring BESS file descriptor.
exportfdReq	*integer*	Number of export ring read indication.
exportfdReqFail	*integer*	Number of export ring read indication failure.
fcudPoolAvailable	*integer*	Number of BESS flow-cache user data available.
fcudPoolFull	*integer*	Number of BESS flow-cache user data pool allocation failure.
fcudPoolGet	*integer*	Number of BESS flow-cache user data pool allocation.
fcudPoolInuse	*integer*	Number of BESS flow-cache user data in use.
fcudPoolPut	*integer*	Number of BESS flow-cache user data pool deallocation.
fcudPoolSize	*integer*	Size of BESS flow-cache user data pool.
flowAdd	*integer*	Number flow added.
flowDel	*integer*	Number flow deleted.
flowKeyUdInReq	*integer*	Number of user data requested.
flowKeyUdInRsp	*integer*	Number of user data response received.

flowKeyUdInRspInvalidFlow **	*integer*	Number of user data response for stale flows. **Since: 4.26.2
flowKeyUdInRspInvalidFt	*integer*	Number of user data response for invalid flow tracker.
flowKeyUdInRspInvalidIntf	*integer*	Number of user data response for invalid flow tracker interface.
flowKeyUdInRspNoFlow	*integer*	Number of user data response for invalid flow.
flowKeyUdInRspNoModule	*integer*	Number of user data response for invalid flow tracker module.
flowKeyUdInRspNoPktSeen	*integer*	Number of user data response with no update.
flowKeyUdInRspNotFound	*integer*	Number of user data response not found.
flowUdReq	*integer*	Number of user data GRPC request sent.
flowUdRsp	*integer*	Number of user data GRPC response received.
getExportInfo	*integer*	Number of BESS get export info requests.
getflowUd	*integer*	Number of BESS get user data requests.
grpcErr	*integer*	Number of errors in GRPC processing.

2.1356 show platform sfe ip route [...]

gates	*table(string) = AsyncGate* see 2.1356.1	Output Gates keyed by interface string.
vrfs	*table(string) = AsyncPrefix* see 2.1356.2	Maps vrf to prefix.

2.1356.1 AsyncGate

outputGate	*integer*	Output gate.
vrfName	*string*	VRF associated with this interface.

2.1356.2 AsyncPrefix

routes	*table(string) = AsyncFec* see 2.1356.2.1	Maps prefix to FEC info.

2.1356.2.1 AsyncFec

fecKey	*integer*	FecKey associated with L2Adj.
nhLocalIndexes	*list of AsyncNhLocalIndex* see 2.1356.2.1.1	nexthop local indexes.

2.1356.2.1.1 AsyncNhLocalIndex

nhLocalIndex	*integer*	Next Hop Local Index.

2.1357 show platform sfe ip route diff

cModule	*ModuleTable* see 2.1357.2	control Module FIBV4 Table.
diffFIB	*boolean*	Diff FIBV4 packet Module with control Module.
diffModuleFec	*ModuleTable* see 2.1357.2	FIBV4 (FEC) Table Diff.
fecs	*table(numeric string) = FibFecInfo* see 2.1357.2.2	Maps a FecKey index to FecInfo.
gates	*table(string) = Gates* see 2.1357.1	Output Gates keyed by interface string.
pModVrfs	*table(string) = PModStatusTable* see 2.1357.3	pModule VRFS.
pModuleFec	*ModuleTable* see 2.1357.2	packet Module FEC Table Diff.
pModuleFib	*ModuleTable* see 2.1357.2	packet Module FIBV4 Table Dump.
routes	*table(string) = Fec* see 2.1357.4	Maps vrf/prefix to its FecKey.

2.1357.1 Gates

outputGate	*integer*	Output gate.

2.1357.2 ModuleTable

fecIndexes	*table(numeric string) = FibFecInfo* see 2.1357.2.2	Maps a FecKey index to FibFecInfo.
fibs	*list of Fib* see 2.1357.2.1	Lists of Fib.
nhInfos	*list of NextHop* see 2.1357.2.3	Lists of NextHop.

2.1357.2.1 Fib

fecKey	*integer*	FecKey associated with L2Adj.
nextHops	*string*	Comma separated list of Next Hop(s).
prefix	*string*	Prefix.
vrfName	*string*	VrfName.

2.1357.2.2 FibFecInfo

fecInfos	*list of pModNextHop* see 2.1357.2.2.1	Lists of pModNextHop.
numberOfUniqueNextHops	*integer*	Number of Unique Next Hops.

2.1357.2.2.1 pModNextHop

gate	*integer*	pModule Gate.
nhLocalIndex	*integer*	Next Hop Local Index.

2.1357.2.3 NextHop

ethAddr	*MAC address*	Destination MAC address.
intf	*interface name*	Egress interface.
nhLocalIndex	*integer*	Next Hop Local Index.
[pktCmd]	*enum: pktCmdTrapVxlanULFlood | pktCmdDrop | pktCmdForwardAndTrap | pktCmdReceive | pktCmdForward | pktCmdSlowReceive | pktCmdNone | pktCmdArpTrap*	Next Hop Type.

2.1357.3 PModStatusTable

activeStatus	*boolean*	pModule status.

2.1357.4 Fec

fecKey	*integer*	FecKey associated with L2Adj.

2.1358 show platform sfe ip route multicast

routes	*table(string) = McastGates* see 2.1358.1	Maps vrf/prefix to its ogate.

2.1358.1 McastGates

cpuGate	*integer*	ogate used for multicast control packet.
dataGate	*integer*	ogate used for multicast data packet.

2.1359 show platform sfe ipv6 route [...]

see 2.1356

2.1360 show platform sfe l3 next-hops [...]

fecs	*table(numeric string) = AsyncFecInfo* see 2.1360.1	Maps a FecKey index to FecInfo.

2.1360.1 AsyncFecInfo

nhInfos	*list of AsyncNextHop* see 2.1360.1.1	Lists of NextHops in this fec.
numberOfUniqueNextHops	*integer*	Number of Unique Next Hops.

2.1360.1.1 AsyncNextHop

l2Adj	*integer*	L2Adj from fecToL2Adjs.
nhFecIndex	*integer*	Index of this nh in the NextHopKey hashmap.
[nhGate] **	*integer*	Next Hop Gate. **Since: 4.28.0
[nhIndex] **	*integer*	Next Hop Index. **Since: 4.28.0
nhIntf	*interface name*	Egress interface.
nhLocalIndex	*integer*	Next Hop Local Index.
nhMac	*MAC address*	Destination MAC address.
[nhType] **	*enum: ecmp | single*	Next Hop Type. **Since: 4.28.0
[nhVlanId] **	*integer*	Next Hop VLAN ID. **Since: 4.28.0
[packetCmd]	*enum: pktCmdTrapVxlanULFlood | pktCmdDrop | pktCmdForwardAndTrap | pktCmdReceive | pktCmdForward | pktCmdSlowReceive | pktCmdNone | pktCmdArpTrap*	Next Hop Type.

2.1361 show platform sfe l3 next-hops diff

cModule	*ModuleTable* see 2.1357.2	control Module Next Hop Table.
diffModule	*ModuleTable* see 2.1357.2	Next Hop Table Diff.
diffNH	*boolean*	Diff Next Hop packet Module with control Module.
fecToL2Adjs	*table(numeric string) = string*	Maps a FecKey to L2Adj.
fecs	*table(numeric string) = FecInfo* see 2.1361.1	Maps a FecKey index to FecInfo.
l2Adjs	*table(string) = Fec* see 2.1357.4	Maps a L2Adj index to FecKey.
pModIntfs	*table(Interface name) = PModStatusTable* see 2.1357.3	pModule interfaces.
pModule	*ModuleTable* see 2.1357.2	packet Module Next Hop Table.

2.1361.1 FecInfo

fecInfos	*list of NextHop* see 2.1357.2.3	Lists of NextHop.
numberOfUniqueNextHops	*integer*	Number of Unique Next Hops.

2.1362 show platform sfe l3 peer-group

peerGroupIdToVtepIpMap	*table(numeric string) = string*	Map for peer group id to vtepIp.

2.1363 show platform sfe l3 software summary

adjCreates	*integer*	Number of adjacency creates.
adjDeletes	*integer*	Number of adjacency deletes.
adjUpdates	*integer*	Number of adjacency updates.
ipVersion	*enum: IPV6 | IPV4*	Version of IP.
lpmCreates	*integer*	Number of LPM creates.
lpmDeletes	*integer*	Number of LPM deletes.
lpmUpdates	*integer*	Number of LPM updates.
lpmUpdatesFecChanged	*integer*	Number of LPM updates where Fec Changed.
lpmUpdatesFecUpdated	*integer*	Number of LPM updates where Fec Updated.
routeDeletes	*integer*	Number of routes deleted.
routeInserts	*integer*	Number of routes inserted.
routeUpdates	*integer*	Number of routes updated.
routeUpdatesFecChanged	*integer*	Number of route updates where Fec Changed.
routeUpdatesFecUpdated	*integer*	Number of route updates where Fec Updated.
routeUpdatesIgnored	*integer*	Number of route updates ignored.

2.1364 show platform sfe l3 unicast summary

forwardingMode	*enum: storeAndForward | defaultMode | cutThrough*	Forwarding mode.
lpmCreateV4 **	*integer*	Number of IPV4 LPM creates. **Since: 4.21.3
lpmCreateV6 **	*integer*	Number of IPV6 LPM creates. **Since: **currentRelease**
lpmDeleteV4 **	*integer*	Number of IPV4 LPM deletes. **Since: 4.21.3
lpmDeleteV6 **	*integer*	Number of IPV6 LPM deletes. **Since: **currentRelease**
lpmRoutes	*integer*	Number of IPV4 and IPV6 LPM routes.
lpmRoutesV4	*integer*	Number of IPV4 LPM routes.
lpmRoutesV6 **	*integer*	Number of IPV6 LPM routes. **Since: **currentRelease**
lpmUpdateV4 **	*integer*	Number of IPV4 LPM updates. **Since: 4.21.3
lpmUpdateV6 **	*integer*	Number of IPV6 LPM updates. **Since: **currentRelease**
mcastEnabledV4	*boolean*	Status of IPV4 multicast routing.
mplsEnabled	*boolean*	Status of MPLS routing.
platformAdjSharing	*boolean*	Status of Platform Adjacency sharing.
routeDeletionDelay	*floating point number*	Route deletion delay.
ucastEnabledV4	*boolean*	Status of IPV4 unicast routing.
ucastEnabledV6	*boolean*	Status of IPV6 unicast routing.

2.1365 show platform sfe licensing

tcLicenseCreated	*boolean*	License TC has been created.
throttledInterfaces	*table(Interface name) = integer*	A mapping of interface to its throttled speed in Mbps.

2.1366 show platform sfe mirroring

sessions	*table(string) = MirrorSession* see 2.1366.1	Map from session name to session details.

2.1366.1 MirrorSession

globalId	*integer*	Global Id.
mirrorGroup	*integer*	Mirror group.
sfeModules	*table(string) = SfeModule* see 2.1366.1.1	Map from module name to module details.
srcIntfs	*table(Interface name) = SourceIntf* see 2.1366.1.2	Map from source interface to direction.
tgtIntfs	*table(Interface name) = TargetIntf* see 2.1366.1.3	Map from target interface to encapsulation type.

2.1366.1.1 SfeModule

sfeCounters	*table(string) = SfeSuccessFailCounter* see 2.1366.1.1.2	Counter for success and failure.
sfeSourceIntfs	*table(Interface name) = SfeSourceIntf* see 2.1366.1.1.1	Map from source interface to output gate.
sfeTargetIntfs	*table(Interface name) = SfeTargetIntf* see 2.1366.1.1.3	Map from target interface to input gate.

2.1366.1.1.1 SfeSourceIntf

direction	*enum: Tx | Both | Rx*	direction of traffic to be mirrored.
gate	*integer*	Output gate number.

2.1366.1.1.2 SfeSuccessFailCounter

fail	*integer*	Failure counter.
success	*integer*	Success counter.

2.1366.1.1.3 SfeTargetIntf

gate	*integer*	Input gate number.

2.1366.1.2 SourceIntf

direction	*enum: Tx | Both | Rx*	direction of traffic to be mirrored.

2.1366.1.3 TargetIntf

encapType	*string*	Encapsulation type.

2.1367 show platform sfe mroute AF [...]

groups	*table(IPv4 or IPv6 address) = MfibMrouteGroup* see 2.1367.1	A mapping of group address to MFIB mroute groups.

2.1367.1 MfibMrouteGroup

mroutes	*table(IPv4 or IPv6 address) = MfibMroute* see 2.1367.1.1	A mapping of source address to MFIB mroutes.

2.1367.1.1 MfibMroute

iifRegister	*boolean*	Route is iif register.
[incomingInterface]	*interface name*	Incoming interface for the route.
[incomingInterfaceFrr]	*interface name*	Incoming interface for fast failover route.
[notifyInterface]	*interface name*	Notify interface for the route.
oifRegister	*boolean*	Route is oif register.
outgoingInterfaceGates	*table(Interface name) = OutgoingInterfaceGate* see 2.1367.1.1.1	A mapping of outgoing interface to gate information.
[staticFastdropInterface]	*interface name*	Interface with static fastdrop installed.
toCpu	*boolean*	Route has local listeners.

2.1367.1.1.1 OutgoingInterfaceGate

outgoingInterface	*interface name*	Outgoing interface for the route.
outgoingInterfaceGate	*integer*	Software networking gate for outgoing interface.

2.1368 show platform sfe nat counters

dyDstAddAttempt	*integer*	Number of dynamic destination add attempts.
dyDstAddSuccess	*integer*	Number of dynamic destination add successes.
dyDstData	*integer*	Number of dynamic destination records.
dyDstDelAttempt	*integer*	Number of dynamic destination delete attempts.
dyDstDelSuccess	*integer*	Number of dynamic destination delete successes.
dyDstModuleFlows	*integer*	Number of dynamic destination module flows.
dyDstTrap	*integer*	Number of dynamic destination trap records.
dyDstTrapAddAttempt	*integer*	Number of dynamic destination trap add attempts.
dyDstTrapAddSuccess	*integer*	Number of dynamic destination trap add successes.
dyDstTrapDelAttempt	*integer*	Number of dynamic destination trap delete attempts.
dyDstTrapDelSuccess	*integer*	Number of dynamic destination trap delete successes.
dyDstUpdAttempt	*integer*	Number of dynamic destination update attempts.
dyDstUpdSuccess	*integer*	Number of dynamic destination update successes.
dySrcAddAttempt	*integer*	Number of dynamic source add attempts.
dySrcAddSuccess	*integer*	Number of dynamic source add successes.
dySrcData	*integer*	Number of dynamic source records.
dySrcDelAttempt	*integer*	Number of dynamic source delete attempts.
dySrcDelSuccess	*integer*	Number of dynamic source delete successes.
dySrcModuleFlows	*integer*	Number of dynamic source module flows.
dySrcTrap	*integer*	Number of dynamic source trap records.
dySrcTrapAddAttempt	*integer*	Number of dynamic source trap add attempts.
dySrcTrapAddSuccess	*integer*	Number of dynamic source trap add successes.
dySrcTrapDelAttempt	*integer*	Number of dynamic source trap delete attempts.
dySrcTrapDelSuccess	*integer*	Number of dynamic source trap delete successes.
dySrcUpdAttempt	*integer*	Number of dynamic source update attempts.
dySrcUpdSuccess	*integer*	Number of dynamic source update successes.
modFlowProcessingFail	*integer*	Number of failure in dynamic flow processing by module.
modProcessingError	*integer*	Number of errors in processing grpc messages by module.

stDstAclData	*integer*	Number of static destination ACL records.
stDstAddAttempt	*integer*	Number of static destination add attempts.
stDstAddSuccess	*integer*	Number of static destination add successes.
stDstData	*integer*	Number of static destination NAT records.
stDstDelAttempt	*integer*	Number of static destination delete attempts.
stDstDelSuccess	*integer*	Number of static destination delete successes.
stDstModuleFlows	*integer*	Number of static destination module flows.
stDstUpdAttempt	*integer*	Number of static destination update attempts.
stDstUpdSuccess	*integer*	Number of static destination update successes.
stSrcAclData	*integer*	Number of static source ACL records.
stSrcAddAttempt	*integer*	Number of static source add attempts.
stSrcAddSuccess	*integer*	Number of static source add successes.
stSrcData	*integer*	Number of static source records.
stSrcDelAttempt	*integer*	Number of static source delete attempts.
stSrcDelSuccess	*integer*	Number of static source delete successes.
stSrcModuleFlows	*integer*	Number of static source module flows.
stSrcUpdAttempt	*integer*	Number of static source update attempts.
stSrcUpdSuccess	*integer*	Number of static source update successes.
twData **	*integer*	Number of twice records. **Since: 4.27.0
twDstAddAttempt **	*integer*	Number of twice destination add attempts. **Since: 4.27.0
twDstAddSuccess **	*integer*	Number of twice destination add successes. **Since: 4.27.0
twDstDelAttempt **	*integer*	Number of twice destination delete attempts. **Since: 4.27.0
twDstDelSuccess **	*integer*	Number of twice destination delete successes. **Since: 4.27.0
twDstModuleFlows **	*integer*	Number of twice destination module flows. **Since: 4.27.0
twDstUpdAttempt **	*integer*	Number of twice destination update attempts. **Since: 4.27.0
twDstUpdSuccess **	*integer*	Number of twice destination update successes. **Since: 4.27.0
twSrcAddAttempt **	*integer*	Number of twice source add attempts. **Since: 4.27.0
twSrcAddSuccess **	*integer*	Number of twice source add successes. **Since: 4.27.0
twSrcDelAttempt **	*integer*	Number of twice source delete attempts. **Since: 4.27.0
twSrcDelSuccess **	*integer*	Number of twice source delete successes. **Since: 4.27.0

`twSrcModuleFlows` **	*integer*	Number of twice source module flows. **Since: 4.27.0
`twSrcUpdAttempt` **	*integer*	Number of twice source update attempts. **Since: 4.27.0
`twSrcUpdSuccess` **	*integer*	Number of twice source update successes. **Since: 4.27.0
`twTrapData` **	*integer*	Number of twice ACL records. **Since: 4.27.0

2.1369 show platform sfe nat module pool-mgr

`[flowCacheEntryNotFound]` **	*integer*	FlowCache entry not found. **Since: 4.28.1
`[natAddressSelectionHash]` **	*boolean*	Address allocation algorithm hash enabled. **Since: 4.29.1
`[natDropRule]` **	*boolean*	Drop rule enabled. **Since: **currentRelease**
`[natDropRuleFullCone]` **	*boolean*	Drop rule full cone enabled. **Since: **currentRelease**
`[natDropRuleSymmetric]` **	*boolean*	Drop rule symmetric enabled. **Since: **currentRelease**
`[natFlowAddErr]` **	*integer*	Erros during inserting into NAT flow table. **Since: 4.30.1
`[natFlowCacheEexistsErr]` **	*integer*	Errors inserting flowCache entry, entry exists. **Since: 4.29.0
`[natFlowCacheFwdErr]` **	*integer*	Errors inserting flowCache entry for NAT flow, forward flow. **Since: 4.29.0
`[natFlowCacheRvsErr]` **	*integer*	Errors inserting flowCache entry for NAT flow, reverse flow. **Since: 4.29.0
`[natFlowCacheUserObjExistsErr]` **	*integer*	Errors inserting flowCache entry, entry with user object exists. **Since: **currentRelease**
`[natFlowExportAddAllocErr]` **	*integer*	Allocation errors during flow export add operation. **Since: 4.29.1
`[natFlowExportDelAllocErr]` **	*integer*	Allocation errors during flow export delete operation. **Since: 4.29.1
`[natFlowExportUpdateAllocErr]` **	*integer*	Allocation errors during flow export update operation. **Since: 4.29.1
`[natFlowFwdAddErr]` **	*integer*	Erros during the insertion of the forward entry into NAT flow table. **Since: **currentRelease**
`[natFlowRvsAddErr]` **	*integer*	Erros during the insertion of the reverse entry into NAT flow table. **Since: **currentRelease**
`[natIpAddressTranslationAny]` **	*boolean*	Address allocation IP translation any enabled. **Since: 4.29.1
`[natObjectNotFound]` **	*integer*	User object not found in flowCache entry. **Since: 4.28.1
`[natRestartInProgress]` **	*boolean*	Hitless restart in progress. **Since: **currentRelease**
`[natSourceIpOnlyHash]` **	*boolean*	Address allocation source IP only hash enabled. **Since: 4.29.1

[numDropRuleErrors] **	*integer*	Address allocate failure, drop rule is set to true. **Since: **currentRelease**
[numDropRuleFullConeErrors] **	*integer*	Address allocate failure, drop rule full cone is set to true. **Since: **currentRelease**
[numDropRuleSymmetric-Errors] **	*integer*	Address allocate failure, drop rule symmetric is set to true. **Since: **currentRelease**
[numFlowCacheAllocErrors] **	*integer*	Errors allocating from ArNatFlowCacheEntry mempool. **Since: 4.30.0
[numFlowCacheDeleteErrors] **	*integer*	Errors deleting from flow cache. **Since: 4.30.0
[numFlowCacheEntries]	*integer*	Entries currently in flowCacheTable.
[numFlowCacheEntriesAlloc] **	*integer*	Entries allocated in flowCacheTable. **Since: 4.30.1
[numFlowCacheEntriesFree] **	*integer*	Entries freed in flowCacheTable. **Since: 4.30.1
[numFlowCacheErrors]	*integer*	Failures inserting flowCache entry.
[numNatFlowEntries]	*integer*	Entries in natFlowTable.
[numRestartInProgress-Errors] **	*integer*	Address allocate failure, restart in progress. **Since: **currentRelease**
[numRwInfo]	*integer*	Rewrite information currently allocated.
[numRwInfoAlloc] **	*integer*	Rewrite information allocated. **Since: 4.30.1
[numRwInfoFree] **	*integer*	Rewrite information freed. **Since: 4.30.1
[numTwiceRwInfo]	*integer*	Rewrite information currently allocated for TwiceNat.
[numTwiceRwInfoAlloc] **	*integer*	Rewrite information allocated for TwiceNat. **Since: 4.30.1
[numTwiceRwInfoFree] **	*integer*	Rewrite information freed for TwiceNat. **Since: 4.30.1
[poolAddrNotAvail] **	*integer*	Address allocate failure, address not available. **Since: 4.29.1
[poolAttempts]	*integer*	Address allocate attempts.
[poolFailure]	*integer*	Address allocate failure.
[poolFlowCacheFwdErr]	*integer*	Errors inserting flowCache entry, forward flow.
[poolFlowCacheRvsErr]	*integer*	Errors inserting flowCache entry, reverse flow.
[poolFlowExportErr] **	*integer*	Errors in exporting flow information. **Since: 4.29.1
[poolLookupHits]	*integer*	Lookup hits, address in-use.
[poolMissing]	*integer*	Address allocate failure, missing poolId.
[poolRwInfoAllocErr] **	*integer*	Errors allocating from ArNatRwInfo mempool. **Since: 4.28.1

[poolSuccess]	*integer*	Address allocate success.
[poolTotal]	*integer*	Address allocate function calls.
[poolTwiceRwInfoAllocErr] **	*integer*	Errors allocating from ArNatTwiceRwInfo mempool. **Since: 4.28.1

2.1370 show platform sfe nat pool [...]

detail	*boolean*	Verbose output.
poolRanges	*list of PoolRange* see 2.1370.1	NAT pool ranges configured in the platform.

2.1370.1 PoolRange

addrEnd	*IPv4 address*	NAT pool range start IP address.
addrLast	*IPv4 address*	Last IP address allocated in the NAT pool range.
addrStart	*IPv4 address*	NAT pool range start IP address.
entryId	*integer*	NAT pool range hardware entry ID.
nextEntryId	*integer*	Next NAT pool range hardware entry ID.
overload **	*boolean*	NAT overload pool. **Since: 4.26.2
poolId	*integer*	NAT pool ID.
portEnd	*integer*	NAT pool range start L4 port.
portLast	*integer*	Last L4 port allocated in the NAT pool range.
portStart	*integer*	NAT pool range start L4 port.

2.1371 show platform sfe qos

cpuReceiveLimit	*integer*	Control plane receive limit.
rateUnit	*enum: rateUnitInvalid* | *rateUnitbps* | *rateUnitKbps* | *rateUnitPps* | *rateUnitMbps*	Rate unit of Control plane receive limit.
txModeCpu	*enum: txModeEnqueue* | *txModeAdaptive* | *txModeDirect* | *txModeInvalid*	Tx Mode CPU.
txModeInterface	*enum: txModeEnqueue* | *txModeAdaptive* | *txModeDirect* | *txModeInvalid*	Tx Mode Interface.

2.1372 show platform sfe route-cache ip hitbit (vrf VRF)

hitbitEntries	*list of HitbitEntry* see 2.1372.1	List of hitbit entries.

2.1372.1 HitbitEntry

hitbit	*integer*	Hitbit value.
prefix	*IP address prefix*	IP prefix.

2.1373 show platform sfe route-cache ip hitbit summary

vrfs	*table(string) = integer*	Mapping of VRF to hitbit count.

2.1374 show platform sfe route-cache summary

revision 2

hitbitCount	*integer*	Number of IPv4 prefixes in hitbit table.
ipv4NotificationFailCounter	*integer*	IPv4 prefix notification fail count.
ipv4NotificationSuccess-Counter	*integer*	IPv4 prefix notification success count.
ipv4PrefixesAgedOutCount	*integer*	Number of IPv4 prefixes aged out.
ipv4PrefixesMovedToHwCount	*integer*	Number of IPv4 prefixes moved to HW.
ipv4RoutePrefixHwSpaceLeft	*integer*	Space left in HW for IPv4 prefixes.
kickAttemptCount	*integer*	Kick attempt count in BESS.
[kickFailCount]	*integer*	Kick attempt fail count in BESS.
prefixEnqFailCount	*integer*	Prefix enqueue fail count in BESS.
readKickNotificationCount	*integer*	Read kick notification count.
routeCachePrefixReqCount	*integer*	Route cache prefix request count.

2.1375 show platform sfe status

bessBuffers	*integer*	Number of bess buffers.
cloudInstanceType	*string*	Cloud instance type.
cpuGhz	*floating point number*	Clock speed of CPU in Ghz.
cpuModel	*string*	Processor model name.
dpMem	*integer*	Memory for datapath in MB.
dpdkVersion **	*string*	DPDK version. **Since: 4.29.1
isolCpuHigh	*integer*	Upper limit of the range of isolated CPUs.
isolCpuLow	*integer*	Lower limit of the range of isolated CPUs.
lastReloadTime	*string*	Last Sfe agent reload time.
maxVrfs	*integer*	Maximum number of VRFs.
nCpCores	*integer*	Number of control plane cores.
nDpCores	*integer*	Number of datapath cores.
nicInfo	*table(Interface name) = NicInfo* see 2.1375.1	Dictionary of NIC information keyed by interface name.
num1GbHugepages	*integer*	Number of 1GB hugepages allocated.
num2MbHugepages	*integer*	Number of 2MB hugepages allocated.
platform	*string*	Cloud platform.
threadsPerCore	*integer*	Number of threads per core.
vrfCount	*integer*	Number of VRFs in use.

2.1375.1 NicInfo

busInfo	*string*	PCI bus information.
driver	*string*	Driver.
l3SubIntfCapable	*boolean*	Layer-3 sub-interface capable.
macAddr	*MAC address*	MAC address.
nicType	*enum: physical | nicUnknown | acceleratedNetworking | vf | vSwitch | passthrough*	NIC Type.
pciAddr	*string*	PCI address.
version	*string*	Version.

2.1376 show platform smbus counters

devices	*table(string) = DeviceModel* see 2.1376.1	Mapping from SMBus device address to device counter info.

2.1376.1 DeviceModel

name	*string*	Name of SMBus device.
rxByteCount	*integer*	Number of rx bytes processed.
rxEstByteRate	*floating point number*	Estimated rate of rx bytes per second.
txByteCount	*integer*	Number of tx bytes processed.

txEstByteRate	*floating point number*	Estimated rate of tx bytes per second.

2.1377 show platform smbus counters errors

devices	*table(string) = DeviceErrorModel* see 2.1377.1	Mapping from SMBus device address to device error counter info.

2.1377.1 DeviceErrorModel

ack	*integer*	Number of ack errors for SMBus device.
busConflict	*integer*	Number of bus conflict errors for SMBus device.
name	*string*	Name of SMBus Device.
timeout	*integer*	Number of timeout errors for SMBus device.

2.1378 show platform tfa pbr nexthops

nexthopsInUse	*list of ArfaPbrNexthopModel* see 2.1378.1	list of nexthops in use.

2.1378.1 ArfaPbrNexthopModel

nexthop	*IP address*	nexthop in use.
vrf	*string*	vrf in which the nexthop is used.

2.1379 show platform trident ((counters raw) | (counters raw (interface INTERFACE) | (all (interface INTERFACE)))) [...]

interfaces	*table(Interface name) = PlatformCounterInterfaceRawModel* see 2.1379.1	Mapping between port interface and interface registers and tables.

2.1379.1 PlatformCounterInterfaceRawModel

registers	*table(string) = PlatformCounterUnitRawModel* see 2.1379.1.1.1	Mapping between register name and register information.
tables	*table(string) = PlatformCounterTableRawModel* see 2.1379.1.1	Mapping between table name and table information.

2.1379.1.1 PlatformCounterTableRawModel

elements	*table(string) = PlatformCounterUnitRawModel* see 2.1379.1.1.1	Mapping between table index and table information.

2.1379.1.1.1 PlatformCounterUnitRawModel

[byteCount]	*integer*	Absolute byte count.
[count]	*integer*	Absolute count (register-specific).
[packetCount]	*integer*	Absolute packet count.

2.1380 show platform trident agent [...]

agents	*table(string) = TridentAgentModel* see 2.1380.1	Agent information by chip name.

2.1380.1 TridentAgentModel

chipName	*string*	Chip name.
configEnabled	*boolean*	Agent config enabled.
configGenId	*integer*	Agent config GenID.
fapId	*integer*	Fap ID.
logicalFapId	*integer*	Logical Fap ID.
sliceName	*string*	Slice name.
statusEnabled	*boolean*	Agent status enabled.
statusGenId	*integer*	Agent status GenID.

2.1381 show platform trident agent port [...]

chips	*table(string) = ChipPortsModel* see 2.1381.1	Per-chip port information keyed by chip name.

2.1381.1 ChipPortsModel

interfaces	*table(Interface name) = PortInfoModel* see 2.1381.1.1	Port information keyed by interface.

2.1381.1.1 PortInfoModel

logical	*integer*	Logical port number.
mmu	*integer*	MMU port number.
modId	*integer*	Module ID.
physical	*integer*	Physical port number.

2.1382 show platform trident agent role [...]

agentRoles	*table(string) = RoleModel* see 2.1382.1	Agent role information, keyed by chip name.

2.1382.1 RoleModel

operationalMode	*string*	Slice Operational Mode.
role	*string*	Agent role.
sliceName	*string*	Slice name.

2.1383 show platform trident ces connection [...]

switchcards	*table(string) = ConnectionsModel* see 2.1383.1	Per-switchard CES connections.

2.1383.1 ConnectionsModel

interfaces	*table(Interface name) = ConnectionModel* see 2.1383.1.1	CES connections by keyed by interface.

2.1383.1.1 ConnectionModel

channelId	*integer*	Channel ID.
linecard	*interface name*	Linecard Fabric Interface.
logicalPort	*integer*	Switchcard Logical Port number.
modId	*integer*	Switchcard ModId.
subport	*integer*	Switchcard Subport number.
switchcard	*interface name*	Switchcard Fabric Interface.

2.1384 show platform trident counters [...]

revision 2

cpu **	*table(Interface name) = PlatformCountType-CountersModel* see 2.1384.1	CPU Port Interface Counters. **Since: 4.29.1
cpuMappings **	*table(Interface name) = string*	Chip names for cpu port ids. **Since: 4.29.1
ethernet	*table(Interface name) = PlatformCountType-CountersModel* see 2.1384.1	Front Panel Interface Counters.
fabric	*table(Interface name) = PlatformCountType-CountersModel* see 2.1384.1	Fabric Port Interface Counters.

2.1384.1 PlatformCountTypeCountersModel

count	*PlatformInterfaceCounter-Model* see 2.1384.1.1	Various platform counters for this interface.

2.1384.1.1 PlatformInterfaceCounterModel

drop	*PlatformDropCounter-Model* see 2.1384.1.1.1	Drop platform counters for this interface.
[dropreason] **	*PlatformDropReason-CounterModel* see 2.1384.1.1.2	Drop reason counters for this interface. **Since: 4.28.1
error	*PlatformErrorCounter-Model* see 2.1384.1.1.3	Error platform counters for this interface.
ok	*PlatformOkCounterModel* see 2.1384.1.1.4	Non Error/Drop platform counters for this interface.
sampled	*PlatformSampledCounter-Model* see 2.1384.1.1.5	Sampled counters for this interface.

2.1384.1.1.1 PlatformDropCounterModel

[inDiscards]	*integer*	RX Discards.
[ipv4L3Discard]	*integer*	RX - IPV4 L3 Discard.
[ipv6L3Discard]	*integer*	RX - IPV6 L3 Discard.
[nonCongestionDiscard]	*integer*	RX - Non congestion discards.
[outDiscards]	*integer*	TX Discards.
[rxAccessPortTrillDiscard]	*integer*	RX - Access Port Trill Packets Discarded.
[rxBufferPoolDiscard]	*integer*	RX - IBP discard and CPB full.
[rxFpDrop]	*integer*	RX - Packets dropped by FP.
[rxIngressNFDrop]	*integer*	RX - Dropped as ingress not forwarding.

[rxMCDrop]	*integer*	RX - Receive multicast drop.
[rxMmuDrop]	*integer*	RX - Packets dropped by the MMU.
[rxNetworkPortNonTrill-Discard]	*integer*	RX - Network Port Non-Trill Packets Discarded.
[rxPipelineDrop]	*integer*	RX - Packets dropped by the pipeline.
[rxPolicyDiscard]	*integer*	RX - Receive policy discard.
[rxUrpfDrop]	*integer*	RX - Receive uRPF drop.
[txAccessPortTrillDiscard]	*integer*	TX - Access Port Trill Packets Discarded.
[txL2MCDrop]	*integer*	TX - L2 multicast drop.
[txL3UCAgedDrop]	*integer*	TX - IPV4 L3 unicast aged and dropped packets.
[txMmuDrop]	*integer*	TX - Packets dropped by the MMU.
[txMmuDropRed]	*integer*	TX - Red packets dropped by the MMU.
[txMmuDropYellow]	*integer*	TX - Yellow packets dropped by the MMU.
[txNetworkPortNonTrill-Discard]	*integer*	TX - Network Port Non-Trill Packets Discarded.
[txPipelineDrop]	*integer*	TX - Packets dropped by the pipeline.
[txTTLDrop]	*integer*	TX - Dropped because TTL counter.
[wredDropPktCounter]	*integer*	TX - WRED drop.

2.1384.1.1.2 PlatformDropReasonCounterModel

[CMLFlags]	*integer*	RX - drop due to CML flags.
[DSTFP]	*integer*	RX - drop due to destination FP.
[EMFT]	*integer*	RX - drop due to EM FT.
[IFP]	*integer*	RX - drop due to IFP.
[IFPMeter]	*integer*	RX - drop due to IFP Meter.
[IPMCL3IIFOrRPAIDCheck-Failed]	*integer*	RX - drop due to IPMC L3 IIF.
[IVXLT]	*integer*	RX - drop due to IVXLT.
[PVLANDrop]	*integer*	RX - drop due to PVLAN drop.
[VFP]	*integer*	RX - drop due to VFP.
[blockMaskDrop]	*integer*	RX - drop due to block mask drop.
[dstLinkDown]	*integer*	RX - drop due to dst linkdown.
[dstPBMZero]	*integer*	RX - drop due to dst PBM zero.
[egrMembershipCheckFailed]	*integer*	RX - drop due to egress membership.
[egrSpanningTreeCheck-Failed]	*integer*	RX - drop due to egress spanning tree.
[forteIpv4Gateway] **	*integer*	RX - drop due to FORTE IPV4 gateway. **Since: 4.29.0
[incomingPvlanCheckFailed]	*integer*	RX - drop due to incoming PVLAN.
[ingRecircleHdrDrop] **	*integer*	RX - drop due to EP NIH packet. **Since: 4.29.0
[l2DstDiscard]	*integer*	RX - drop due to L2 dst discard.
[l2DstLookupMiss]	*integer*	RX - drop due to L2 dst lookup.
[l2SrcDiscard]	*integer*	RX - drop due to L2 src discard.
[l2SrcStaticMove]	*integer*	RX - drop due to L2 src static.
[l3DstDiscard]	*integer*	RX - drop due to L3 dst discard.
[l3DstLookupMiss]	*integer*	RX - drop due to L3 dst lookup.
[l3HdrError]	*integer*	RX - drop due to L3 header error.

[l3IIFEqL3OIF]	*integer*	RX - drop due to L3 IIF equals L3.
[l3MtuCheckFailed]	*integer*	RX - drop due to L3 MTU check.
[l3TTLError]	*integer*	RX - drop due to L3 TTL error.
[lagFailoverPortDown]	*integer*	RX - drop due to lag failover port.
[macSaMulticast]	*integer*	RX - drop due to MacSa Multicast.
[membershipCheckFailed]	*integer*	RX - drop due to membership check.
[mplsCtrlPktDrop]	*integer*	RX - drop due to mpls ctrl pkt.
[mplsEcnError]	*integer*	RX - drop due to mpls ECN error.
[mplsLabelActionInvalid]	*integer*	RX - drop due to mpls label action.
[mplsProtectionDrop]	*integer*	RX - drop due to mpls protection.
[mplsReservedLabelExposed]	*integer*	RX - drop due to mpls reserved.
[mplsTTLError]	*integer*	RX - drop due to mpls TTL error.
[mplsTermPolicySelectTable-Drop]	*integer*	RX - drop due to mpls term policy.
[noDrop]	*integer*	Dropped with no reason.
[outerTpidCheckFailed]	*integer*	RX - drop due to outer TPID check.
[pktIntegrityCheckFailed]	*integer*	RX - drop due to packet integrity.
[protocolPktDrop]	*integer*	RX - drop due to protocol packet.
[seqNumCheckFailed]	*integer*	RX - drop due to seq num check.
[spanningTreeCheckFailed]	*integer*	RX - drop due to spanning tree.
[splitHorizonCheckFailed]	*integer*	RX - drop due to split horizon.
[srcPortKnockoutDrop]	*integer*	RX - drop due to source port knockout.
[stormControlDrop]	*integer*	RX - drop due to storm control.
[tunnelAdaptDrop]	*integer*	RX - drop due to tunnel adapt drop.
[tunnelObjectValidation-Failed]	*integer*	RX - drop due to tunnel object.
[tunnelShimHdrError]	*integer*	RX - drop due to tunnel shim.
[tunnelTTLCheckFailed]	*integer*	RX - drop due to tunnel TTL check.
[unknownDropReason] **	*integer*	RX - drop for unknown reason. **Since: 4.29.0
[urpfCheckFailed]	*integer*	RX - drop due to Urpf check failed.

2.1384.1.1.3 PlatformErrorCounterModel

[alignmentErrors]	*integer*	Alignment Errors.
[fcsErrors]	*integer*	FCS Errors.
[frameTooLongs]	*integer*	Frame Too Long.
[frameTooShorts]	*integer*	Undersize packets.
[inErrors]	*integer*	RX Errors.
[internalMacReceiveErrors]	*integer*	RX Internal MAC Errors.
[ipv4L3HeaderError]	*integer*	RX - IPV4 header error.
[ipv6L3HeaderError]	*integer*	RX - IPV6 header error.
[outErrors]	*integer*	TX Errors.
[rxL2MTUError]	*integer*	RX - L2 MTU oversize dropped packets.
[rxTunnelError]	*integer*	RX - Tunnel error packets.
[rxVlanDrop]	*integer*	RX - Receive VLAN drop.
[sqeTestErrors]	*integer*	SQE Test Errors.
[symbolErrors]	*integer*	Symbol Errors.
[txFCSError]	*integer*	TX - FCS error.

[txInvalidVlan]	*integer*	TX - Invalid VLAN.
[txL2MTUError]	*integer*	TX - L2 MTU oversize dropped packets.
[txMACError]	*integer*	TX - Transmit MAC error.
[txPCError]	*integer*	TX - Purge or cell error.
[txSplitHorizonDrop]	*integer*	TX - Packets dropped by Split Horizon.
[txUnknownDrop]	*integer*	TX - Packet dropped due to any condition.
[txVxltMiss]	*integer*	TX - Miss in VXLT table counter.

2.1384.1.1.4 PlatformOkCounterModel

[deferredTransmissions]	*integer*	Deferred Transmissions.
[ecnMarkedPackets]	*integer*	TX - ECN marked packets.
[fragments]	*integer*	Fragments.
[in1024To1518OctetFrames]	*integer*	RX counter bin 1024-1518 bytes.
[in128To255OctetFrames]	*integer*	RX counter bin 128-255 bytes.
[in1519To2047OctetFrames]	*integer*	RX counter bin 1519-2047 bytes.
[in2048To4095OctetFrames]	*integer*	RX counter bin 2048-4095 bytes.
[in256To511OctetFrames]	*integer*	RX counter bin 256-511 bytes.
[in4096To9216OctetFrames]	*integer*	RX counter bin 4096-9216 bytes.
[in512To1023OctetFrames]	*integer*	RX counter bin 512-1023 bytes.
[in64OctetFrames]	*integer*	RX counter bin 64 bytes.
[in65To127OctetFrames]	*integer*	RX counter bin 65-127 bytes.
[in9217To16383OctetFrames]	*integer*	RX counter bin 9217-16383 bytes.
[inBroadcastPkts]	*integer*	RX Broadcast packets.
[inMulticastPkts]	*integer*	RX Multicast packets.
[inOctets]	*integer*	RX bytes.
[inPauseFrames]	*integer*	RX Pause Frames.
[inPfc0Frames]	*integer*	Pfc0 Indications.
[inPfc1Frames]	*integer*	Pfc1 Indications.
[inPfc2Frames]	*integer*	Pfc2 Indications.
[inPfc3Frames]	*integer*	Pfc3 Indications.
[inPfc4Frames]	*integer*	Pfc4 Indications.
[inPfc5Frames]	*integer*	Pfc5 Indications.
[inPfc6Frames]	*integer*	Pfc6 Indications.
[inPfc7Frames]	*integer*	Pfc7 Indications.
[inPfcFrames]	*integer*	Pfc Indications.
[inTotalPkts]	*integer*	RX packets.
[inUcastPkts]	*integer*	RX Unicast.
[ipv4L3Mcast]	*integer*	RX - Routed IPV4 multicast.
[ipv4L3Ok]	*integer*	RX - Good IPV4 L3 Unicast.
[ipv6L3Mcast]	*integer*	RX - Routed IPV6 multicast.
[ipv6L3Ok]	*integer*	RX - Good IPV6 L3 Unicast.
[jabbers]	*integer*	Jabbers.
[out1024To1518OctetFrames]	*integer*	TX counter bin 1024-1518 bytes.
[out128To255OctetFrames]	*integer*	TX counter bin 128-255 bytes.
[out1519To2047OctetFrames]	*integer*	TX counter bin 1519-2047 bytes.
[out2048To4095OctetFrames]	*integer*	TX counter bin 2048-4095 bytes.
[out256To511OctetFrames]	*integer*	TX counter bin 256-511 bytes.
[out4096To9216OctetFrames]	*integer*	TX counter bin 4096-9216 bytes.

[out512To1023OctetFrames]	*integer*	TX counter bin 512-1023 bytes.
[out64OctetFrames]	*integer*	TX counter bin 64 bytes.
[out65To127OctetFrames]	*integer*	TX counter bin 65-127 bytes.
[out9217To16383OctetFrames]	*integer*	TX counter bin 9217-16383 bytes.
[outBroadcastPkts]	*integer*	TX Broadcast packets.
[outMulticastPkts]	*integer*	TX Multicast packets.
[outOctets]	*integer*	TX bytes.
[outPauseFrames]	*integer*	TX Pause Frames.
[outPfc0Frames]	*integer*	TX Pfc0 Requests.
[outPfc1Frames]	*integer*	TX Pfc1 Requests.
[outPfc2Frames]	*integer*	TX Pfc2 Requests.
[outPfc3Frames]	*integer*	TX Pfc3 Requests.
[outPfc4Frames]	*integer*	TX Pfc4 Requests.
[outPfc5Frames]	*integer*	TX Pfc5 Requests.
[outPfc6Frames]	*integer*	TX Pfc6 Requests.
[outPfc7Frames]	*integer*	TX Pfc7 Requests.
[outPfcFrames]	*integer*	TX Pfc Requests.
[outUcastPkts]	*integer*	TX Unicast packets.
[rxUC]	*integer*	RX - Good L2, L3 unicast.
[txIpv4L3MCOk]	*integer*	TX - Good IPV4 L3 multicast packets.
[txIpv4L3UCOk]	*integer*	TX - Good IPV4 L3 unicast packets.
[txIpv6L3MCOk]	*integer*	TX - Good IPV6 L3 multicast packets.
[txIpv6L3UCOk]	*integer*	TX - Good IPV6 L3 unicast packets.
[wredEctDropPktCounter]	*integer*	TX - WRED ECT drop.
[wredNonEctDropPktCounter]	*integer*	TX - WRED non-ECT drop.

2.1384.1.1.5 PlatformSampledCounterModel

[adapt_2_miss_drop]	*integer*	TX - Adapt 2 miss drop (sampled).
[classbasedmovedrop]	*integer*	RX - Class based station movement drop (sampled).
[dsfrag]	*integer*	RX - DOS fragment error packets (sampled).
[dsicmp]	*integer*	RX - DOS ICMP error packets (sampled).
[dsl3he]	*integer*	RX - DOS L3 header error packets (sampled).
[dsl4he]	*integer*	RX - DOS L4 header error packets (sampled).
[dstdiscarddrop]	*integer*	RX - L2/L3 lookup DST_DISCARD drop (sampled).
[egressProtectionDataDrop]	*integer*	TX - Protection data drop (sampled).
[fcmPortClass2RxDiscards]	*integer*	RX - FCOE L2 RX discarded frames (sampled).
[fcmPortClass2RxFrames]	*integer*	RX - FCOE L2 RX frames (sampled).
[fcmPortClass2TxFrames]	*integer*	TX - FCOE L2 TX frames (sampled).
[fcmPortClass3RxDiscards]	*integer*	RX - FCOE L3 RX discarded frames (sampled).
[fcmPortClass3RxFrames]	*integer*	RX - FCOE L3 RX frames (sampled).
[fcmPortClass3TxFrames]	*integer*	TX - FCOE L3 TX frames (sampled).
[hGoE]	*integer*	RX - HGoE Rx frames (sampled).

[hghdre]	*integer*	RX - Higig header error packets (sampled).
[higigLoopbackDrop]	*integer*	RX - HiGig loopback drop (sampled).
[higigUnknownOpcodDrop]	*integer*	RX - HiGig unknown opcode drop (sampled).
[imbp]	*integer*	RX - Bridged multicast packets (sampled).
[imirror]	*integer*	RX - HiGig mirror packet (sampled).
[imrp4]	*integer*	RX - Routed IPv4 multicast packets (sampled).
[imrp6]	*integer*	RX - Routed IPv6 multicast packet (sampled).
[ingressProtectionDataDrop]	*integer*	RX - Protection data drop (sampled).
[irhol]	*integer*	RX - HiGig End-to-End HOL rx packets (sampled).
[iribp]	*integer*	RX - HiGig End-to-End IBP rx packets (sampled).
[irpse]	*integer*	RX - HiGig IPIC pause rx (sampled).
[iunkhdr]	*integer*	RX - Unknown HigGig header type packet (sampled).
[laglup]	*integer*	RX - LAG failover loopback packets (sampled).
[laglupd]	*integer*	RX - LAG backup port down (sampled).
[maclmt_drop]	*integer*	RX - Mac limit exceeded and packet dropped (sampled).
[maclmt_nodrop]	*integer*	RX - Mac limit exceeded and packet not dropped (sampled).
[macsaequalmacda]	*integer*	RX - Dos Attack L2 Packets (same MAC source and destination) (sampled).
[mcidxe]	*integer*	RX - Multicast index error packets (sampled).
[modidtoolargedrop]	*integer*	TX - MODID greater than 31 drop counter (sampled).
[mtuerr]	*integer*	RX - Packets trapped to CPU due to egress L3 MTU violation (sampled).
[nonHGoE]	*integer*	RX - Non HGoE Rx frames on an HGoE port (sampled).
[parityd]	*integer*	RX - Parity error packets (sampled).
[pdisc]	*integer*	RX - Policy discard (sampled).
[pfm]	*integer*	RX - PFM multicast drop (sampled).
[pktmodtoolargedrop]	*integer*	TX - Byte additions too large drop counter (sampled).
[rdisc]	*integer*	RX - IBP discard and CBP full (sampled).
[rdrop]	*integer*	RX - Port bitmap zero drop condition (sampled).
[rfildr]	*integer*	RX - Packets dropped by FP (sampled).
[rhgmc]	*integer*	RX - HiGig lookup MC cases (sampled).
[rhguc]	*integer*	RX - HiGig lookup UC cases (sampled).
[rimdr]	*integer*	RX - Multicast (L2 + L3) packets that are dropped (sampled).
[ripc4]	*integer*	RX - Good IPv4 packets (sampled).
[ripc6]	*integer*	RX - Good IPv6 packets (sampled).
[ripd4]	*integer*	RX - IPv4 L3 discard packets (sampled).

[ripd6]	*integer*	RX - IPv6 L3 discard packets (sampled).
[riphe4]	*integer*	RX - IPv4 header error packets (sampled).
[riphe6]	*integer*	RX - IPv6 header error packets (sampled).
[rportd]	*integer*	RX - Packets dropped when ingress port is not in forwarding state (sampled).
[rtun]	*integer*	RX - Number of tunnel packets received (sampled).
[rtune]	*integer*	RX - Tunnel error packets (sampled).
[ruc]	*integer*	RX - Good unicast (L2+L3 packets) (sampled).
[rxBFDUnknownAchError]	*integer*	RX - BFD packets with invalid ACH header (sampled).
[rxBFDUnknownControlFrames]	*integer*	RX - Unrecognized BFD control packets (sampled).
[rxBFDUnknownVersion-Discards]	*integer*	RX - Unknown BFD version or discard dropped (sampled).
[rxBlockMaskDrop]	*integer*	RX - Bitmap of ports blocked (sampled).
[rxBpduDrop]	*integer*	RX - BPDU packets dropped (sampled).
[rxCompositeError]	*integer*	RX - HW related error like lookup table with parity error (sampled).
[rxDSL2HE]	*integer*	RX - Dos Attack L2 Packets (sampled).
[rxDlbRhResolutionError]	*integer*	RX - DLB or RH resolution error (sampled).
[rxEcmpResolutionError]	*integer*	RX - Packets dropped due to ECMP resolution error (sampled).
[rxEgressMaskDrop]	*integer*	RX - Packets dropped due to egress mask (sampled).
[rxEgressMaskSrcPortDrop]	*integer*	RX - Packets dropped due to block traffic from egress based on ingress port (sampled).
[rxEgressVlanMember-Mismatch]	*integer*	RX - Packets dropped due to mismatched egress port for the VLAN (sampled).
[rxFcoeSrcBindError]	*integer*	RX - FCOE source bind check failure (sampled).
[rxFcoeSrcFpmaError]	*integer*	RX - Source FPMA prefix check failure (sampled).
[rxFcoeVftDrop]	*integer*	RX - FCOE source bind check failure (sampled).
[rxFcoeZoneError]	*integer*	RX - FCOE_ZONE lookup miss (sampled).
[rxFcoeZoneLookupMiss]	*integer*	RX - FCOE_ZONE lookup miss (sampled).
[rxFieldChangeL2NoRedirect-Drop]	*integer*	RX - Change L2 fields without deploying redirect action drop (sampled).
[rxFieldDelayDrop]	*integer*	RX - Packets dropped due to ingress FP delayed action (sampled).
[rxFieldRedirectMaskDrop]	*integer*	RX - Packets dropped due to FP redirection mask (sampled).
[rxINTVectorMismatch]	*integer*	RX - INT packets dropped due to request vector match failure (sampled).
[rxIpInIpDrop]	*integer*	RX - IPinIP tunnel process failure (sampled).

[rxIpmcDrop]	*integer*	RX - IPMC process failure (sampled).
[rxL2DstDiscardDrop]	*integer*	RX - L2 DST_DISCARD drop (sampled).
[rxL2HeaderError]	*integer*	RX - L2 header error (sampled).
[rxL2RpfError]	*integer*	RX - Incoming port/lag/svp check failure (sampled).
[rxL2SrcDiscardDrop]	*integer*	RX - L2 SRC_DISCARD drop (sampled).
[rxL2SrcStaticMoveDrop]	*integer*	RX - Port movement of static L2 addr (sampled).
[rxL2TtlError]	*integer*	RX - L2 ZONE TTL check failure (sampled).
[rxL3DisableDrop]	*integer*	RX - V4/V6 L3_ENABLE unset drop (sampled).
[rxL3DstDiscardDrop]	*integer*	RX - L3 DST_DISCARD drop (sampled).
[rxL3DstLookupDrop]	*integer*	RX - IPMC processing drop, or L3_DESTINATION == 0 for L3UC packets (sampled).
[rxL3HeaderError]	*integer*	RX - L3 header error (sampled).
[rxL3SrcDiscardDrop]	*integer*	RX - L3 SRC_DISCARD drop (sampled).
[rxL3TtlError]	*integer*	RX - L3 TTL check failure (sampled).
[rxL3TunnelLookupMiss]	*integer*	RX - L3 Tunnel lookup miss (sampled).
[rxLAGFailLoopback]	*integer*	RX - LAG fail loopback packets in packet discard stage (sampled).
[rxLinkDownDrop]	*integer*	RX - packets dropped due to link down or destination port disabled (sampled).
[rxMCError]	*integer*	RX - Packets dropped due to multicast errors (sampled).
[rxMacBlockMaskDrop]	*integer*	RX - Packets dropped due to MAC block mask (sampled).
[rxMacControlDrop]	*integer*	RX - Packets dropped due to MAC control frame (sampled).
[rxMacIpBindError]	*integer*	RX - MAC to IP bind check failure (sampled).
[rxMacSaEqual0Drop]	*integer*	RX - Packets dropped due to MACSA == 0 (sampled).
[rxMimDrop]	*integer*	RX - MiM tunnel process failure (sampled).
[rxMplsActionInvalid]	*integer*	RX - MPLS invalid action (sampled).
[rxMplsControlWordInvalid]	*integer*	RX - MPLS invalid control word (sampled).
[rxMplsEntropyDrop]	*integer*	RX - MPLS Entropy label in unallowed range (sampled).
[rxMplsErr]	*integer*	RX - MPLS packets with error (sampled).
[rxMplsGalNotBosDrop]	*integer*	RX - MPLS Generic Label is not BOS (sampled).
[rxMplsLabelLookupMiss]	*integer*	RX - MPLS label lookup miss (sampled).
[rxMplsLookupMiss]	*integer*	RX - MPLS lookup miss (sampled).
[rxMplsPayloadInvalid]	*integer*	RX - MPLS invalid payload (sampled).
[rxMplsSeqNumberError]	*integer*	RX - MPLS control word sequence number error (sampled).
[rxMyStationDrop]	*integer*	RX - MY_STATION.DISCARD drop (sampled).

[rxNatDrop]	*integer*	RX - NAT process failure (sampled).
[rxNextHopDrop]	*integer*	RX - NEXT_HOP drop (sampled).
[rxNextHopOrL3EifRange-Error]	*integer*	RX - Packets dropped due to next hop or L3 egress interface ID range error (sampled).
[rxNonUnicastLAGMaskDrop]	*integer*	RX - Packets dropped due to multicast and broadcast trunk block mask (sampled).
[rxNonUnicastMaskDrop]	*integer*	RX - Packets dropped due to block masks (sampled).
[rxPartialLinkDownDrop]	*integer*	RX - non-UC packets dropped partially due to link down or dest port disabled (sampled).
[rxPrivateVlanMismatch]	*integer*	RX - PVLAN VID mismatch (sampled).
[rxPrivateVlanVpFilterDrop]	*integer*	RX - Packets dropped based on src/dst VP type (sampled).
[rxProtocolDrop]	*integer*	RX - Protocol packets dropped (sampled).
[rxProtocolErrorIP4]	*integer*	RX - Packets dropped due to IPv4 protocol error (sampled).
[rxProtocolErrorIP6]	*integer*	RX - Packets dropped due to IPv6 protocol error (sampled).
[rxSrcKnockoutDrop]	*integer*	RX - SGPP == DGPP or SVP == DVP, etc (sampled).
[rxSrcRouteDrop]	*integer*	RX - MACSA is multicast(bit 40 == 1) (sampled).
[rxStgDrop]	*integer*	RX - STP not in forwarding state (sampled).
[rxStgMaskDrop]	*integer*	RX - Packets dropped due to spanning tree group state (sampled).
[rxStormControlDrop]	*integer*	RX - Storm control metering drop (sampled).
[rxTagUntagDiscardDrop]	*integer*	RX - PORT_DIS_UNTAG/PORT_DIS_TAG drop (sampled).
[rxTimeSyncDrop]	*integer*	RX - Network time sync packets dropped (sampled).
[rxTpidMismatch]	*integer*	RX - VLAN TPID mismatch (sampled).
[rxTunnelDecapEcnError]	*integer*	RX - Tunnel decap ECN error (sampled).
[rxTunnelErr]	*integer*	RX - Packets dropped due to tunnel errors (sampled).
[rxTunnelInterfaceError]	*integer*	RX - IPMC tunnel incoming interface is incorrect (sampled).
[rxTunnelObjectValidation-Fail]	*integer*	RX - Tunnel termination failure (sampled).
[rxTunnelShimError]	*integer*	RX - Tunnel shim header error (sampled).
[rxTunnelTtlError]	*integer*	RX - Tunnel TTL check failure (sampled).
[rxUnknownMacSaDrop]	*integer*	RX - Unknown MACSA packets dropped (sampled).
[rxVfiAdaptLookupMiss]	*integer*	RX - Packets dropped due a lookup miss drop during ingress VFI adaption (sampled).
[rxVfiP2pDrop]	*integer*	RX - Packets arrived from a VP that is not programmed in PT2PT connection (sampled).

[rxVlanBlockMaskDrop]	*integer*	RX - Packets dropped due to VLAN blocked ports (sampled).
[rxVlanCrossConnectOrNon-UnicastDrop]	*integer*	RX - L2 forwarding lookup miss or PBT non unicast packets drop (sampled).
[rxVlanMemberMismatch]	*integer*	RX - Ingress port not in VLAN membership (sampled).
[rxVlanMismatch]	*integer*	RX - VLAN ID invalid (sampled).
[rxVlanTranslate1Lookup-Miss]	*integer*	RX - VLAN Translate1 lookup miss (sampled).
[rxVlanTranslate2Lookup-Miss]	*integer*	RX - LAN Translate2 lookup miss (sampled).
[rxVxlanDecapSrcIpLookup-Miss]	*integer*	RX - VXLAN packet dropped due to source IP lookup miss during tunnel decap (sampled).
[rxVxlanTunnelErrorDrop]	*integer*	RX - Drop due to Vxlan tunnel error (sampled).
[rxVxlanVnidLookupMissDrop]	*integer*	RX - Drop due to Vxlan network id lookup miss (sampled).
[taged]	*integer*	TX - Packets dropped due to packet aged (sampled).
[tcfid]	*integer*	TX - Packets dropped when CFI set & packet is untagged or L3 switched for IPMC (sampled).
[tgip4]	*integer*	TX - Good IPv4 L3 UC packets (sampled).
[tgip6]	*integer*	TX - Good IPv6 L3 UC packets (sampled).
[tgipmc4]	*integer*	TX - Good IPv4 IPMC packets (sampled).
[tgipmc6]	*integer*	TX - Good IPv6 IPMC packets (sampled).
[thgi]	*integer*	TX - Unknown HiGig drop (sampled).
[thgmc]	*integer*	TX - HiGig lookedup L3MC packets (sampled).
[thguc]	*integer*	TX - HiGig lookedup L3UC packets (sampled).
[thigig2]	*integer*	TX - Unknown HiGig2 drop (sampled).
[tip_len_fail]	*integer*	TX - IP length check fail drop (sampled).
[tipd4]	*integer*	TX - IPv4 L3 UC Aged and Drop packets (sampled).
[tipd6]	*integer*	TX - IPv6 L3 UC Aged and Drop Packets (sampled).
[tipmcd4]	*integer*	TX - Good IPv4 IPMC packets (sampled).
[tipmcd6]	*integer*	TX - IPv6 IPMC Aged and Drop packets (sampled).
[tl2_mtu]	*integer*	TX - L2 MTU fail drop (sampled).
[tl2mcd]	*integer*	TX - L2 MC packet drop counter (sampled).
[tmirr]	*integer*	TX - mirrored packets (sampled).
[tparity_err]	*integer*	TX - Parity Error drop (sampled).
[tpktd]	*integer*	TX - Packets dropped due to any condition (sampled).
[tsipl]	*integer*	TX - SIP Link Local Drop (sampled).
[tslld]	*integer*	TX - ESIP link local (sampled).
[tstgd]	*integer*	TX - Packets dropped due to spanning tree state not in forwarding state (sampled).

[ttnl]	*integer*	TX - Tunnel packets (sampled).
[ttnle]	*integer*	TX - Tunnel error packets (sampled).
[tttld]	*integer*	TX - Packets dropped due to TTL threshold counter (sampled).
[tvlan]	*integer*	TX - VLAN tagged packets (sampled).
[tvland]	*integer*	TX - Pkts dropped due to invalid VLAN counter (sampled).
[tvxltmd]	*integer*	TX - Packets dropped due to miss in VXLT table counter (sampled).
[txAdaptLookupMiss]	*integer*	TX - packets dropped due to egress adapt lookup miss (sampled).
[txCellTooSmall]	*integer*	TX - Multi-cell packets with SOP cell size <64 bytes after decap (sampled).
[txEgressFilterDrop]	*integer*	TX - Packets dropped by EFP (sampled).
[txEgressPortDrop]	*integer*	TX - EGR_LPORT_PROFILE.DROP (sampled).
[txFwdDomainNotFoundDrop]	*integer*	TX - Forwarding domain not found (sampled).
[txIpmcL2SrcPortPruneDrop]	*integer*	TX - IPMC L2 self port drop (sampled).
[txL3EgressOrIntfRangeError]	*integer*	TX - packets dropped due to L3 next hop and interface indices do not comply with configured allocation bitmap (sampled).
[txLoopbackToLoopbackDrop]	*integer*	TX - Packets dropped due to loopback packet dest is also loopback port (sampled).
[txMmuPurgeDrop]	*integer*	TX - Packets dropped due to MMU purge (sampled).
[txNonUnicastPruneDrop]	*integer*	TX - Non unicast pruning (sampled).
[txNotFwdDomainMemberDrop]	*integer*	TX - Not forwarding domain member (sampled).
[txSplitHorizonCheckDrop]	*integer*	TX - Split horizon check failure (sampled).
[txStgDisableDrop]	*integer*	TX - Packets dropped due to STG disabled (sampled).
[txSvpEsDrop]	*integer*	TX - pkts dropped due to ethernet segment filtering on source virtual port (sampled).
[txSvpRemoveDrop]	*integer*	TX - Virtual port pruning (sampled).
[txVisibilityDrop]	*integer*	TX - Visibility packets dropped (sampled).
[txVplagPruneDrop]	*integer*	TX - VPLAG pruning (sampled).
[urpf]	*integer*	RX - Unicast RPF drop case (sampled).
[vfpdr]	*integer*	RX - VLAN FP drop case (sampled).
[vlandr]	*integer*	RX - VLAN drops (sampled).

2.1385 show platform trident counters fast-polling [...]

averageFetchTime	*floating point number*	Average number of milliseconds to collect counters.
[detail] **	*FastPollInfoDetail* see 2.1385.1	Late poll details. **Since: 4.25.1
lastPollTimestamp	*floating point number*	Timestamp of at the end of last poll interval.
latePollCount	*integer*	Number of times poll interval has been missed.
maximumFetchTime	*floating point number*	Maximum number of milliseconds to collect counters.
pollFastInterval	*integer*	Configured fast polling period in milliseconds.
reEnqueueCount	*integer*	Number of times update requests saturated Schan.
retryDequeueCount	*integer*	Number of times waited for request completion.
totalFetchCount	*integer*	Number of times the event has occurred.
totalPollCount	*integer*	Numer of times the polling has completed.

2.1385.1 FastPollInfoDetail

latePollHistory	*list of LatePollEvent* see 2.1385.1.2	Last late poll events.
[statsLastDay]	*FastPollStats* see 2.1385.1.1	Statistics over the last day.
[statsLastHour]	*FastPollStats* see 2.1385.1.1	Statistics over the last hour.

2.1385.1.1 FastPollStats

averageFetchTime	*floating point number*	Average time needed to collect counters (ms).
latePollCount	*integer*	Number of times poll interval has been missed.
maximumFetchTime	*floating point number*	Maximum time needed to collect counters (ms).
statsStartTimestamp	*floating point number*	Start of the monitored period.
totalPollCount	*integer*	Numer of times the polling has completed.

2.1385.1.2 LatePollEvent

delay	*floating point number*	Polling delay (in milliseconds).
timestamp	*floating point number*	Timestamp of the late poll.

2.1386 show platform trident counters fast-polling interface [...]

interfaces	*table(Interface name) = InterfaceFastCounters* see 2.487.3	A mapping of interface to its fast counters.

2.1387 show platform trident counters interface (FABRIC | ETH | UETH)

revision 2

see 2.1384

2.1388 show platform trident counters polling

averageFetchTime	*floating point number*	Average number of seconds to collect counters.
lastPollTimestamp	*floating point number*	Timestamp of at the end of last poll interval.
latePollCount	*integer*	Number of times poll interval has been missed.
maximumFetchTime	*floating point number*	Maximum number of seconds to collect counters.
pollInterval	*floating point number*	Configured polling period in seconds.
reEnqueueCount	*integer*	Number of times update requests saturated Schan.
retryDequeueCount	*integer*	Number of times waited for request completion.
totalFetchCount	*integer*	Number of times the event has occurred.
totalPollCount	*integer*	Numer of times the polling has completed.

2.1389 show platform trident counters sampling

sampledCounterChip	*list of SampledCounterChipModel* see 2.1389.1	Sampling information per chip.

2.1389.1 SampledCounterChipModel

chip	*list of string*	Chips this sampling information applies to.
rxSampledData	*SampledData* see 2.1389.1.1	Rx pipeline sampeld data.
txSampledData	*SampledData* see 2.1389.1.1	Tx pipeline sampeld data.

2.1389.1.1 SampledData

samplingRegisters	*integer*	Number of pipeline sampling registers.
samplingTriggers	*integer*	Number of triggers being sampled.

`triggerActiveRatio`	*floating point number*	Percentage of the time a given trigger is being counted.

2.1390 show platform trident directflow [...]

slices	*table(string) = StrataSliceInfo* see 2.1390.1	A mapping from slice name to flow.

2.1390.1 StrataSliceInfo

flows	*table(string) = DirectFlowFlowName* see 2.1390.1.1	A mapping from flow name to resource allocation info.

2.1390.1.1 DirectFlowFlowName

fabricResources	*table(string) = FabricResourceInfo* see 2.1390.1.1.2	A mapping of fabric name to fabric tcam state.
hgClassId	*integer*	HiGig class ID.
hgClassResourceState	*enum: Acquired | Pending | NA*	HiGig class resource state.
ipmcIndex **	*integer*	L3_IPMC index. **Since: 4.22.0
ipmcIndexResourceState **	*enum: Acquired | Pending | NA*	IPMC resource state. **Since: 4.22.0
l2McIndex	*integer*	L2MC index.
l2McResourceState	*enum: Acquired | Pending | NA*	L2MC resource state.
mirrorResources	*table(string) = FlowMirrorTargetIntf* see 2.1390.1.1.3	A mapping of mirrorKey to mirror resource info.
nextHopIndex	*integer*	Nexthop index.
nextHopResourceState	*enum: Acquired | Pending | NA*	NextHop resource state.
opaqueId **	*integer*	Opaque object ID. **Since: 4.22.0
opaqueIdResourceState **	*enum: Acquired | Pending | NA*	Opaque object resource state. **Since: 4.22.0
[tcamEntries] **	*table(string) = DirectFlowTcamEntry* see 2.1390.1.1.1	TCAM entries associated with the flow. **Since: 4.23.0
tcamResourceState	*enum: Acquired | Pending*	TCAM resource state.

2.1390.1.1.1 DirectFlowTcamEntry

[entryId]	*integer*	Flow TCAM entry ID.

2.1390.1.1.2 FabricResourceInfo

fabricStatus	*enum: Acquired | Pending*	Fabric index program state.

2.1390.1.1.3 FlowMirrorTargetIntf

direction	*enum: Ingress | Egress*	Mirroring direction.
mirrorIndex	*integer*	Mirroring index.
mirrorResourceState	*enum: Acquired | Pending | NA*	L2MC resource state.
targetIntf	*string*	Mirroring target interface.

2.1391 show platform trident ecmp-lag (software | shadow) [...]

ecmpLags	*table(string) = EcmpLagStatus* see 2.1391.1	A mapping from lag Name to EcmpLag Status.

2.1391.1 EcmpLagStatus

lagId	*integer*	Ecmp Lag Id.
members	*table(numeric string) = EcmpLagMember* see 2.1391.1.1	A mapping from Index to EcmpLagMember.

2.1391.1.1 EcmpLagMember

intfId	*interface name*	Interface Name.
modId	*integer*	Module Id.
portId	*integer*	Port Id.

2.1392 show platform trident egress event debug [...]

chips	*table(string) = EgrEventListModel* see 2.1392.1	Map of chip name to list of egress debug events seen.

2.1392.1 EgrEventListModel

[eventCounters] **	*table(string) = integer*	Map of eress debug events seen since last cleared.Value represents the egress debug event count. **Since: 4.27.1
events	*list of string*	List of egress debug events seen since last cleared.Order is chip-dependent. Each appears at most once per chip.

2.1393 show platform trident exact-match [...]

entries	*table(string) = StrataEmCliModelEntry* see 2.1393.1	EM entries indexed by their ID.

2.1393.1 StrataEmCliModelEntry

appName	*string*	Application name.
[drop]	*boolean*	Packet has been dropped.
[dstIp]	*IP address*	Destination IP address.
[dstL4Port]	*integer*	Destination L4 Port.
[ecmp] **	*boolean*	Nexthop is an ECMP group. **Since: 4.24.2
[extractionCtrlId]	*integer*	Extraction Control Id.
[groupClassId] **	*integer*	Table Id. **Since: 4.24.2
[hiGig] **	*boolean*	Match traffic ingressing on HiGig. **Since: 4.24.2
[hits] **	*integer*	Number of times entry has been hit. **Since: 4.24.2
[ipProtocol]	*integer*	IP Protocol.
[logicalSrcPort] **	*integer*	Incoming Source Port. **Since: 4.25.0
ltId	*integer*	HW Logical Table ID.
[mcastGroupIp] **	*IP address*	VxLAN Underlay Multicast Group IP address. **Since: 4.25.0
[mcastSrcIp] **	*IP address*	VxLAN Underlay Multicast Source IP address. **Since: 4.25.0
[moduleId] **	*integer*	Module ID. **Since: 4.25.0
[nextHopId] **	*integer*	Nexthop value. **Since: 4.24.2
[numAppEntries]	*integer*	Number of Exact Match entries available for this application.
[opaqueId1]	*integer*	Opaque Id 1.
[opaqueId3]	*integer*	Opaque Id 3.
[redirectDvp] **	*integer*	Redirect DVP value. **Since: 4.24.2
[srcIp]	*IP address*	Source IP address.
[srcL4Port]	*integer*	Source L4 Port.
[tableSize]	*integer*	Maximum number of single wide Exact Match entries available.
[vrf] **	*integer*	VRF ID. **Since: 4.26.2

2.1394 show platform trident exact-match summary

entries	*table(string) = StrataEmCliSummaryModelEntry* see 2.1394.1	EM entries indexed by their appName.

2.1394.1 StrataEmCliSummaryModelEntry

[errorCount]	*integer*	Number of Exact Match entries that could not be installed for this application.
[installedEntries]	*integer*	Number of Exact Match entries installed for this application.
[numAppEntries]	*integer*	Number of Exact Match entries available for this application.
[tableSize]	*integer*	Maximum number of single wide Exact Match entries available.

2.1395 show platform trident fabric counters fec [...]

chips	*table(string) = PlatformSliceFecCounterModel* see 2.1395.1	Mapping from chip name to FEC counters.

2.1395.1 PlatformSliceFecCounterModel

fapId **	*integer*	Fabric switch FAP ID. **Since: 4.25.0
modId	*integer*	Fabric switch module ID (legacy attribute, use fapId instead).
ports	*table(Interface name) = InterfaceFecCounterModel* see 2.1395.1.1	Mapping from interface to FEC Counters.

2.1395.1.1 InterfaceFecCounterModel

correctable	*integer*	RX - Correctable FEC error.
uncorrectable	*integer*	RX - Uncorrectable FEC error.

2.1396 show platform trident fabric counters queue [...]

fabricChips	*table(string) = StrataFabricQueueCounterDropModel* see 2.1396.1	Mapping from fabric chip to fabric ports.

2.1396.1 StrataFabricQueueCounterDropModel

fapId **	*integer*	Fabric switch FAP ID. **Since: 4.25.0
modId	*integer*	Fabric switch module ID (legacy attribute, use fapId instead).

ports	*table(Interface name) = StrataInterfaceQueueDropCounters* see 2.1396.1.1	Mapping from fabric port to drop counters.

2.1396.1.1 StrataInterfaceQueueDropCounters

egressMcDroppedPackets	*integer*	Multicast egress packets dropped.
egressUcDroppedPackets	*integer*	Unicast egress packets dropped.
ingressDroppedPackets	*integer*	Ingress packets dropped.

2.1397 show platform trident fabric counters queue detail [...]

fabricChips	*table(string) = StrataFabricInterfaceCounterModel* see 2.1397.1	Mapping from fabric chip to fabric ports.

2.1397.1 StrataFabricInterfaceCounterModel

fapId **	*integer*	Fabric switch FAP ID. **Since: 4.25.0
modId	*integer*	Fabric switch module ID (legacy attribute, use fapId instead).
ports	*table(Interface name) = StrataInterfaceEgressQueueDestinationTypeCounter* see 2.1397.1.1	Mapping from fabric port to egress tx-queue counters.

2.1397.1.1 StrataInterfaceEgressQueueDestinationTypeCounter

destinationQueues	*table(string) = Counters* see 2.261.1.1.1.2	Mapping from destination queue (unicast/multicast) to counter values.

2.1398 show platform trident fabric counters rate

see 2.483

2.1399 show platform trident flexcounters l3intf in summary

l3intfs	*table(string) = L3IntfInFlexCounterInfo* see 2.1399.1	A mapping from l3 intf to L3IntfInFlexCounterInfo.

2.1399.1 L3IntfInFlexCounterInfo

baseCounterIndex	*integer*	Base counter index for flexcounter.
counterIndex	*integer*	Counter index for flexcounter.

offsetMode	*integer*	Offset mode for flexcounter.
poolId	*integer*	Pool id for flexcounter.

2.1400 show platform trident flexcounters l3intf in values

l3intfs	*table(string) = FlexCounterValuesColl* see 2.1400.1	A mapping from l3Intf addresss to flexCounterValues.

2.1400.1 FlexCounterValuesColl

counters	*table(string) = FlexCounterValues* see 2.1400.1.1	a container list for FlexCounterValues in counterTable.
snapShotCounters	*table(string) = FlexCounterValues* see 2.1400.1.1	a container list for FlexCounterValues in snapShotTable.

2.1400.1.1 FlexCounterValues

nbytes	*integer*	Number of bytes for flexcounter.
npackets	*integer*	Number of packets for flexcounter.

2.1401 show platform trident flexcounters l3intf out summary

see 2.1399

2.1402 show platform trident flexcounters l3intf out values

see 2.1400

2.1403 show platform trident flexcounters multicast summary

groups	*table(IPv4 or IPv6 address) = McastSources* see 2.1403.1	A mapping of Group IP addresses to Source IP addresses.

2.1403.1 McastSources

sources	*table(IPv4 or IPv6 address) = Vlans* see 2.1403.1.1	A mapping of Source IP addresses to VLAN IDs.

2.1403.1.1 Vlans

vlans	*table(numeric string) = Vrfs* see 2.1403.1.1.1	A mapping of VLAN IDs to VRFs.

2.1403.1.1.1 Vrfs

vrfs	*table(numeric string) = CounterIndexes* see 2.1403.1.1.1.1	A mapping of VRFs to their Counter Indices.

2.1403.1.1.1.1 CounterIndexes

counterIndexes	*table(numeric string) = McastSummaryTableInfo* see 2.1403.1.1.1.2	A mapping of Counter Indices to their counter information.

2.1403.1.1.1.2 McastSummaryTableInfo

baseCounterIndex	*integer*	Base Counter Index for flexcounter.
offsetMode	*enum: 1 | 0 | 2 | 3*	Offset Mode for flexcounter.
poolId	*integer*	Pool Id for flexcounter.

2.1404 show platform trident flexcounters multicast values

groups	*table(IPv4 or IPv6 address) = McastSources* see 2.1404.1	A mapping of Group IP addresses to Source IP addresses.

2.1404.1 McastSources

sources	*table(IPv4 or IPv6 address) = Vlans* see 2.1404.1.1	A mapping of Source IP addresses to VLAN IDs.

2.1404.1.1 Vlans

vlans	*table(numeric string) = Vrfs* see 2.1404.1.1.1	A mapping of VLAN IDs to VRFs.

2.1404.1.1.1 Vrfs

vrfs	*table(numeric string) = CounterIndexes* see 2.1404.1.1.1.1	A mapping of VRFs to their Counter Indices.

2.1404.1.1.1.1 CounterIndexes

counterIndexes	*table(numeric string) = McastValueTables* see 2.1404.1.1.1.2	A mapping of Counter Indices to their Snapshot and Counter Tables.

2.1404.1.1.1.2 McastValueTables

counterTable	*McastValueTableInfo* see 2.1404.1.1.1.3	Counter Table for flex counter.

snapshotTable	*McastValueTableInfo* see 2.1404.1.1.1.3	Snapshot Table for flex counter.

2.1404.1.1.1.3 McastValueTableInfo

nbytes	*integer*	Byte counter for mroute.
npackets	*integer*	Packet counter for mroute.

2.1405 show platform trident flexcounters requests

revision 2

egressRequests	*table(string) = FlexRequest* see 2.1405.1	Mapping from egress action name to action config.
ingressRequests	*table(string) = FlexRequest* see 2.1405.1	Mapping from ingress action name to action config.

2.1405.1 FlexRequest

[actionId]	*integer*	Action ID.
countOnDrop	*enum: countOnlyDropped | ignoreIfDropped | countAll*	Counter behavior regarding dropped packets.
[countOnMirror]	*boolean*	The counter updates on switch-generated mirror packets (Only applicable to EPIPE instances).
counterMode	*enum: slim | simple | normal | wide*	Mode for dividing counter entries.
index	*ValueSource* see 2.1405.1.1	The operation used to compute the counter index.
operatorA	*enum: setValue | setBit | max | rdecrement | decrement | min | increment | xorOp | noop | average*	Operation used to update counter A.
operatorB	*enum: setValue | setBit | max | rdecrement | decrement | min | increment | xorOp | noop | average*	Operation used to update counter B.
requestName	*string*	Name of the request.
sourceAgent	*string*	Name of the agent that created the request.
sourceTableName	*string*	Name of the source table.
status	*enum: allocated | pending*	Status of the request.
valueA	*ValueSource* see 2.1405.1.1	The source value for updating counter A.
valueB	*ValueSource* see 2.1405.1.1	The source value for updating counter B.

2.1405.1.1 ValueSource

lowerBits	*ObjectSelector* see 2.1405.1.1.1	Information to compute lower bits of op profile.
upperBits	*ObjectSelector* see 2.1405.1.1.1	Information to compute upper bits of op profile.

2.1405.1.1.1 ObjectSelector

maskSize	*integer*	Size of mask in bits (actual size is this value + 1).
objectName	*string*	Value used to compute the index, A, or B.
shiftLeft	*integer*	Bits to shift value left indicated by srcField.

shiftRight	*integer*	Bits to shift value right indicated by srcField.

2.1406 show platform trident flexcounters requests summary

egressRequests	*table(string) = FlexRequestSummary* see 2.1406.1	Egress FlexCounters2 requests, keyed by name.
ingressRequests	*table(string) = FlexRequestSummary* see 2.1406.1	Ingress FlexCounters2 requests, keyed by name.

2.1406.1 FlexRequestSummary

[actionId]	*integer*	Allocated action ID.
counterMode	*enum: slim | simple | normal | wide*	Mode for dividing counter entries.
[pool]	*integer*	Base pool allocated to counter action.
requestedSize	*integer*	Requested number of entries.
[size]	*integer*	Number of entries to allocated from pool.
sourceAgent	*string*	Name of the agent providing the request.
sourceTableName	*string*	Name of the source table.
status	*enum: allocated | pending*	Status of the request.

2.1407 show platform trident flexcounters summary

poolDict	*table(string) = PoolInfoList* see 2.1407.1	Mapping from feature ID to list of pool info.

2.1407.1 PoolInfoList

poolInfoList	*table(numeric string) = PoolInfo* see 2.1407.1.1	Mapping from request ID to pool info.

2.1407.1.1 PoolInfo

poolType	*enum: egress | ingress*	Strata pool type.
[responseData]	*ResponseData* see 2.1407.1.1.1	Model containing corresponding response information.
validRequest	*boolean*	Denotes whether request ID came from a pool request.

2.1407.1.1.1 ResponseData

poolId	*integer*	Pool ID.
size	*integer*	Number of counters.
startIndex	*integer*	Start Index.

2.1408 show platform trident flexcounters vni egress summary

vnis	*table(string) = VniFlexCounterInfo* see 2.1408.1	A mapping from VNIs to VniFlexCounterInfo for egress.

2.1408.1 VniFlexCounterInfo

baseCounterIndex	*integer*	Base counter index for flexcounter.
offsetMode	*integer*	Offset mode for flexcounter.
poolId	*integer*	Pool ID for flexcounter.
vfiIndex	*integer*	VNI index for flexcounter.

2.1409 show platform trident flexcounters vni egress values

vnis	*table(string) = FlexCounterValuesColl* see 2.1400.1	A mapping from Vni addresss to flexCounterValues.

2.1410 show platform trident flexcounters vni ingress summary

vnis	*table(string) = VniFlexCounterInfo* see 2.1408.1	A mapping from VNIs to VniFlexCounterInfo for ingress.

2.1411 show platform trident flexcounters vni ingress values

see 2.1409

2.1412 show platform trident flexcounters vtep decap summary

vteps	*table(string) = VtepDecapFlexCounterInfo* see 2.1412.1	A mapping from remote VTEP IP addresss to VtepDecapFlexCounterInfo.

2.1412.1 VtepDecapFlexCounterInfo

baseCounterIndex	*integer*	Base counter index for flexcounter.
offsetMode	*integer*	Offset mode for flexcounter.
poolId	*integer*	Pool id for flexcounter.
svpIndex	*integer*	Svp Index for flexcounter.

2.1413 show platform trident flexcounters vtep decap values

vteps	*table(IPv4 address) = FlexCounterValuesColl* see 2.1400.1	A mapping from remote VTEP IP addresss to flexCounterValues.

2.1414 show platform trident flexcounters vtep encap summary

vteps	*table(string) = VtepFlexCounterInfo* see 2.1414.1	A mapping from remote VTEP IP addresss to VtepEncapFlexCounterInfo.

2.1414.1 VtepFlexCounterInfo

baseCounterIndex	*integer*	Base counter index for flexcounter.
dvpIndex	*integer*	Hardware Index for flexcounter.
offsetMode	*integer*	Offset mode for flexcounter.
poolId	*integer*	Pool id for flexcounter.

2.1415 show platform trident flexcounters vtep encap values

see 2.1413

2.1416 show platform trident flow tracker collector software

collectors	*table(string) = Collector* see 2.1416.1	Information on collectors keyed by their ID.

2.1416.1 Collector

collState	*enum: programmed | idle | failed | unknown*	State of collector programming.
dscpValue	*integer*	DSCP value.
dstIp	*IP address*	Destination IP (IP address of collector).
dstPort	*integer*	UDP destination port.
ipfixIntfUp **	*boolean*	Status of IPFIX kernel interface. **Since: 4.30.0
maxMsgSize	*integer*	Max message size.
srcIp	*IP address*	Source IP (IP address of exporter).
srcPort	*integer*	UDP source port.
[vrfName] **	*string*	VRF name. **Since: 4.30.0

2.1417 show platform trident flow tracker counters [...]

faps	*table(numeric string) = StftOneLinecardCounter* see 2.1417.1	A mapping of FAP ID to its flow groups.

2.1417.1 StftOneLinecardCounter

flowGroups	*table(numeric string) = StftOneFlowGroupCounter* see 2.1417.1.1	A mapping of flow group ID to its counters.

2.1417.1.1 StftOneFlowGroupCounter

active	*integer*	Active flow count.
ageOut	*integer*	AgeOut flow count.
exceed	*integer*	Exceed flow count.
groupName	*string*	Hardware flow group name.
learned	*integer*	Learned flow count.
missed	*integer*	Missed flow count.

2.1418 show platform trident flow tracker group software

flowgroup	*list of FlowGroup* see 2.1418.1	List of flow groups.

2.1418.1 FlowGroup

collectorId	*integer*	Associated collector ID.
elements	*list of InfoElement* see 2.1418.1.1	Exported information elements.
[encapsulation] **	*list of string*	Packet encapsulation for this group. **Since: 4.25.1
fgId	*integer*	Hardware ID of the flow group.
[fgName] **	*string*	Hardware flow group name. **Since: 4.25.1
flowKey	*list of string*	Packet header fields that flows in this group are keyed on.
flowParam	*list of string*	Tracking parameters.
[mirrorConfig] **	*MirrorConfig* see 2.1418.1.2	Mirror config. **Since: 4.25.1
[seqNo] **	*integer*	Sequence number of the flow group. **Since: 4.25.1
templateId	*integer*	Template ID in IPFIX data record.

2.1418.1.1 InfoElement

eid	*integer*	Element ID.
enterprise	*boolean*	This is an enterprise element.
isKey	*boolean*	This element is a flow key.
length	*integer*	Length of the element.
param	*string*	Name of the element.

2.1418.1.2 MirrorConfig

initialCopy	*integer*	Initial packets to be mirrored.
mirrorIntervalFixed	*integer*	Sampling interval for packets in fixed algorithm.
mirrorIntervalRandom	*integer*	Sampling interval N to sample packets with a probability of 1/N.
mirrorSelectorAlgo	*enum: random | sampleModeNone | fixed*	Sample selection algorithm.
sessionName	*string*	Mirror session name.

2.1419 show platform trident flow tracker summary

revision 2

logicalFaps	*table(numeric string) = OneLinecardFlowTracker* see 2.1419.1	A mapping of Logical FAP ID to flow tracker information on it.

2.1419.1 OneLinecardFlowTracker

flowtrackers	*list of FlowTracker* see 2.1419.1.1	Configured flow trackers.

2.1419.1.1 FlowTracker

collector	*integer*	Collector ID of the flow tracker.
flowgroups	*table(string) = integer*	A mapping of flow group name to its ID.
intfs	*list of Interface name*	List of interfaces on which the flow tracker is enabled.
name	*string*	Name of the flow tracker.
state	*enum: flowTrackerIdle | flowTrackerGroupDirDeleteDone | flowTrackerFtfpProgrammed | flowTrackerFtfpDeleteDone | flowTrackerIntfClassSetupFailed | flowTrackerAny | flowTrackerIntfClassDeleteDone | flowTrackerFailed | flowTrackerProgrammed | flowTrackerIntfClassSetupInProg | flowTrackerGroupDirSetupInProg | flowTrackerNotProgrammed | flowTrackerDeleteDone | flowTrackerFtfpSetupInProg | flowTrackerGroupDirDeleteInProg | flowTrackerFtfpDeleteInProg | flowTrackerFtfpSetupFailed | flowTrackerGroupDirProgrammed | flowTrackerIntfClassProgrammed*	State of the flow tracker.

2.1420 show platform trident forwarding-table partition

chipModel	*string*	Chip Model.
[egressVlanRemapTableSize]	*integer*	Egress VLAN remap table size.
[exactMatchSupported]	*boolean*	Exact Match feature supported.
[l2TableSize]	*integer*	L2 table size.
[l3TableSize]	*integer*	L3 table size.
[lpmTableSize]	*integer*	LPM table size.
[totalEmEntries]	*integer*	Exact Match table size.
[totalL2Entries]	*integer*	L2 table size.

[totalL3Entries]	*integer*	L3 host table size.
uftMode	*enum: flexible | 1 | 2 | 3 | alpm | 0*	UFT Mode.
[urpfEnabled]	*boolean*	URPF enabled.
[vlanRemapTableSize]	*integer*	VLAN remap table size.

2.1421 show platform trident forwarding-table partition flexible

revision 2

[alpmEntries] **	*integer*	Shared ALPM entries. **Since: 4.25.2
emBankMappings	*table(numeric string) = integer*	EM physical to logical bank mapping.
emHashOffsets	*table(numeric string) = integer*	Exact Match hash offsets indexed by physical bank id.
[exactMatchEntries]	*integer*	Shared Exact Match entries.
[exactMatchMaxEntries]	*integer*	Maximum Exact Match UFT entries.
[exactMatchNumHwLogicalTables]	*integer*	Maximum Exact Match hardware logical tables.
[exactMatchNumHwLookups]	*integer*	Maximum Exact Match hardware lookups.
[exactMatchSupported]	*boolean*	Exact Match feature supported.
[flexibleUftEnabled]	*boolean*	Flexible UFT mode is enabled.
[initialized]	*boolean*	Flexible UFT mode attributes initialized.
l2BankMappings	*table(numeric string) = integer*	L2 physical to logical bank mapping.
l2HashOffsets	*table(numeric string) = integer*	L2 hash offsets indexed by physical bank id.
[l2SharedEntries]	*integer*	Shared L2 entries.
l3BankMappings	*table(numeric string) = integer*	L3 physical to logical bank mapping.
l3HashOffsets	*table(numeric string) = integer*	L3 hash offsets indexed by physical bank id.
[l3SharedEntries]	*integer*	Shared L3 entries.
[numEmSharedBanks]	*integer*	Shared Exact Match UFT banks.
[numL2SharedBanks]	*integer*	Shared L2 UFT banks.
[numL3SharedBanks]	*integer*	Shared L3 UFT banks.
[uftAlpmEntriesPerBank] **	*integer*	ALPM Entries per bank. **Since: 4.25.2
[uftCrcStringBytes]	*integer*	Maximum number of bytes in a bucket.
[uftExactMatchEntriesPerBucket]	*integer*	Exact Match entries per bucket.
[uftKeyShift]	*integer*	Number of bits to shift a key before hashing.
[uftL2EntriesPerBucket]	*integer*	L2 entries per bucket.
[uftL2MinEntries]	*integer*	Minimum L2 UFT entries.
[uftL2NumDedicatedBanks]	*integer*	Minimum L2 UFT banks.
[uftL2SharedMaxEntries]	*integer*	Maximum L2 UFT entries.
[uftL3EntriesPerBucket]	*integer*	L3 entries per bucket.
[uftL3MinEntries]	*integer*	Minimum L3 UFT entries.
[uftL3NumDedicatedBanks]	*integer*	Minimum L3 UFT banks.
[uftL3SharedMaxEntries]	*integer*	Maximum L3 UFT entries.
[uftMaxEntriesPerBucket]	*integer*	Maximum entries per bucket.
[uftMaxSharedBanks]	*integer*	Maximum shared UFT banks.
[uftMaxSharedBuckets]	*integer*	Maximum buckets in shared UFT banks.
[uftMaxSharedEmBanks] **	*integer*	Maximum shared Exact Match UFT banks. **Since: 4.27.1

[uftMaxSharedL2Banks] **	*integer*	Maximum shared L2 UFT banks. **Since: 4.27.1
[uftMaxSharedL3Banks] **	*integer*	Maximum shared L3 UFT banks. **Since: 4.27.1
[uftMode]	*integer*	uftMode.
[uftSharedBankSize]	*integer*	Base entries in shared UFT bank.

2.1422 show platform trident forwardingplane featureagents

agents	*list of AgentModel* see 2.1422.1	List of forwarding plane agents.

2.1422.1 AgentModel

name	*string*	Agent name.

2.1423 show platform trident ingress event debug [...]

chips	*table(string) = IngEventListModel* see 2.1423.1	Map of chip name to list of ingress debug events seen.

2.1423.1 IngEventListModel

[eventCounters] **	*table(string) = integer*	Map of ingress debug events seen since last cleared. Value represents the ingress debug event count. **Since: 4.27.1
events	*list of string*	List of ingress debug events seen since last cleared. Order is chip-dependent. Each appears at most once per chip.

2.1424 show platform trident interface map [...]

chips	*table(string) = ChipEntry* see 2.1424.1	Interface mappings indexed by chip.

2.1424.1 ChipEntry

fapId	*integer*	Fap ID.
interfaces	*table(string) = InterfaceEntry* see 2.1424.1.2	Interface map info indexed by interface or port name.
numPortMax	*integer*	Maximum number of ports.
[openFlowInterfaces]	*table(Interface name) = OpenFlowEntry* see 2.1424.1.1	OpenFlow interface info keyed by interface name.

2.1424.1.1 OpenFlowEntry

modId	*integer*	Module identifier.
portId	*integer*	OpenFlow port identifier.

2.1424.1.2 InterfaceEntry

[egressXpe]	*list of integer*	Egress crosspoint elements.
[ingressXpe]	*list of integer*	Ingress crosspoint elements.
logicalPort	*integer*	Logical port number.
[mmuPort]	*integer*	MMU port number.
[modId]	*integer*	Module identifier.
physicalPort	*integer*	Physical port number.
[pipe]	*integer*	Pipe number.
portName	*string*	Port name.
[vpNativeIndex]	*integer*	Virtual port native index.
[vpTaggedIndex]	*integer*	Virtual port tagged index.

2.1425 show platform trident l2 hardware mac-address-table [...]

chips	*table(string) = MacAddressTable* see 2.1425.1	A mapping of chip name to its MAC address table.

2.1425.1 MacAddressTable

macTableEntries	*list of MacTableEntry* see 2.1425.1.1	List of MAC table entries.

2.1425.1.1 MacTableEntry

classId	*integer*	Class ID.
hwTblIndex **	*integer*	Index in the hardware table. **Since: 4.28.1

interfaces **	*list of string*	Interface for this MAC address. **Since: 4.26.0
macAddress	*MAC address*	MAC address.
macAddressType	*enum: drop | dynamic | dropSrc | static | dropDst*	Type of the MAC address.
[mcastGroupId] **	*integer*	Multicast group ID. **Since: 4.26.0
vlan	*integer*	VLAN ID.

2.1426 show platform trident l2 hardware mac-address-table diff [...]

macTableDiffEntries	*list of MacTableDiffEntry* see 2.1426.1	List of MAC table diff entries.

2.1426.1 MacTableDiffEntry

hwEntries	*table(string) = DiffEntry* see 2.1426.1.1	A mapping of chip name to its hardware entries.
hwTblIndex **	*integer*	Index in the hardware table. **Since: 4.28.1
mac	*MAC address*	MAC address.
swEntries	*list of DiffEntry* see 2.1426.1.1	Software entries.
vlan	*integer*	VLAN ID.

2.1426.1.1 DiffEntry

[classId]	*integer*	Class Id.
[dstDiscard]	*boolean*	Destination Discard.
[invalidEntry]	*boolean*	Invalid Entry.
[macAddressType]	*string*	Type of the MAC address.
[mcastId]	*integer*	Multicast ID.
[mod]	*integer*	Mod.
[port]	*integer*	Port.
[trunkId]	*integer*	Trunk Id.

2.1427 show platform trident l2 hardware mac-address-table missing [...]

macTableMissingEntries	*list of MacTableMissingEntry* see 2.1427.1	List of MAC table missing entries.

2.1427.1 MacTableMissingEntry

hwTblIndex **	*integer*	Index in the hardware table. **Since: 4.28.1
inSw	*boolean*	MAC address present in software.
mac	*MAC address*	MAC address.

modules	*list of string*	Chip cards.
vlan	*integer*	VLAN ID.

2.1428 show platform trident l2 shadow counter [...]

revision 2

cntrTables	*table(string) = CounterTable* see 2.1428.1	Individual Counter tables, indexed by the table name.

2.1428.1 CounterTable

cmds	*table(string) = CounterTableEntry* see 2.1428.1.1	Counters in each HW Table, where the key is the command type inside the table.

2.1428.1.1 CounterTableEntry

deltaAdds	*integer*	Incremental additions on the table.
deltaDeletes	*integer*	Incremental deletions on the table.
deltaModifications	*integer*	Incremental modifications on the table.
totalAdds	*integer*	Total number of add operations on the table.
totalDeletes	*integer*	Total number of delete operations on the table.
totalModifications	*integer*	Total number of modification operations on the table.

2.1429 show platform trident l2 shadow egr-tag-action-profile

egrVlanTagActionProfile-Table	*table(numeric string) = EgrVlanTagActionProfile-Entry* see 2.1429.1	Egress Vlan Tag Action Profile Table.

2.1429.1 EgrVlanTagActionProfileEntry

dtOtagAction	*integer*	Outer VID action for double-taggged packets.
sotOtagAction	*integer*	Outer VID Action for single outer-tagged packets.
utOtagAction	*integer*	Outer VID action for untagged packets.

2.1430 show platform trident l2 shadow egr-vfi

vxlanEgrVfiTable	*table(numeric string) = EgrVfiEntry* see 2.1430.1	Vxlan Egress VFI Table.

2.1430.1 EgrVfiEntry

flexCounterBaseCounterIndex	*integer*	Base Counter Index used to compute the final counter index.
flexCounterOffsetMode	*integer*	Flex Counter Offset Mode to decide the final counter index.
flexCounterPoolNumber	*integer*	Flex Counter Pool Number that will be updated.
ovid	*integer*	Outer VLAN ID field in the entry - this field is valid if the ovidValid field is set to true.
ovidValid	*boolean*	if True, ovid field is valid.
stg	*integer*	Spanning Tree GroupId field in the entry.
stgValid	*boolean*	if True, stg field is valid.
vmcFlag	*integer*	VFI Membership Check (VMC) Enable Flag - this field is valid if vmcFlagValid field is set to true.
vmcFlagValid	*boolean*	if True, vmcFlag (VFI Membership Check) field is valid.
vmpPtr	*integer*	VFI Membership Profile (VMP) Pointer - this field is valid if vmpPtrValid field is set to true.
vmpPtrValid	*boolean*	if True, vmpPtr (VFI Membership Profile Pointer) field is valid.

2.1431 show platform trident l2 shadow egr-vlan-xlate [...]

egrVlanXlateTables	*table(string) = EgrVlanXlateHwTable* see 2.1431.1	Egress Vlan Xlate Tables for all Line Cards in a MultiChip System with the Line Card Name as key.

2.1431.1 EgrVlanXlateHwTable

dvpXlateEntries	*table(numeric string) = EgrVlanXlateDvpEntry* see 2.1431.1.1	Destination Virtual Port based Entries with the key being EntryId which is a hardware Index.
rawXlateEntries	*table(numeric string) = EgrVlanXlateRawEntry* see 2.1431.1.3	All Entries in Raw Format with EntryId as key which is a hardware index. This format is to support the raw option in CLI.
vfiXlateEntries	*table(numeric string) = EgrVlanXlateVfiEntry* see 2.1431.1.4	Virtual Forwarding Instance based Entries with key being EntryId which is a hardware index.
vxlanEncapXlateEntries **	*table(numeric string) = EgrVlanXlateVxlanEncap-Entry* see 2.1431.1.5	Virtual Forwarding Instance based Entries with key being EntryId which is a hardware index. **Since: 4.26.0
xlateEntries	*table(numeric string) = EgrVlanXlateEntry* see 2.1431.1.2	Port Vlan Xlate Entries with the key being EntryId which is a hardware Index.

2.1431.1.1 EgrVlanXlateDvpEntry

dvp	*integer*	Destination Virtual Port.
newOvid	*integer*	New Outer VLAN ID.
ovid	*integer*	Outer VLAN ID.
tagActionProfilePtr	*integer*	Index into Ingress VLAN Tag Action Profile Table.

2.1431.1.2 EgrVlanXlateEntry

newOvid	*integer*	New Outer VLAN ID.
ovid	*integer*	Outer VLAN ID.
portGroupId	*integer*	Port Group ID of the egress Port.
tagActionProfilePtr	*integer*	Index into Ingress VLAN Tag Action Profile Table.

2.1431.1.3 EgrVlanXlateRawEntry

entryValue	*string*	Raw dump of the entry.

2.1431.1.4 EgrVlanXlateVfiEntry

sdTagActionIfPresent	*integer*	Index into Ingress VlanTagActionProfileTable if the packet contains SD Tag.
vfi	*integer*	VFI.
vnId	*integer*	VN ID.

2.1431.1.5 EgrVlanXlateVxlanEncapEntry

newOvid	*integer*	New Outer Dot1q Tag.
ovid	*integer*	Outer VLAN ID.
portGroupId	*integer*	Port Group ID of the egress Port.
tagActionProfilePtr	*integer*	Index into Ingress VLAN Tag Action Profile Table.

2.1432 show platform trident l2 shadow egr-vlan-xlate-2 [...]

egrVlanXlate2Tables	*table(string) = EgrVlanXlate2HwTable* see 2.1432.1	Egress Vlan Xlate 2 Tables keyed by linecards.

2.1432.1 EgrVlanXlate2HwTable

assignIfaMdHdrEntries **	*table(numeric string) = EgrVlanXlate2AssignIfaMdHdrEntry* see 2.1432.1.1	AssignIfaMdHdr entries keyed by their hardware index. **Since: 4.25.1
changeL2FieldsEntries	*table(numeric string) = EgrVlanXlate2L2FieldsEntry* see 2.1432.1.3	L2 Fields keyed by their hardware index.
ifaMdLookup1Entries **	*table(numeric string) = EgrVlanXlate2IfaMdLookup1Entry* see 2.1432.1.2	IfaMdLookup1 entries keyed by their hardware index. **Since: 4.25.1
rawEntries	*table(numeric string) = EgrVlanXlate2RawEntry* see 2.1432.1.4	All Entries in Raw Format keyed by their heardware index.

2.1432.1.1 EgrVlanXlate2AssignIfaMdHdrEntry

actionVector	*integer*	Action vector.
currentLength	*integer*	Current length.
gns	*integer*	Global Namespace.
hopLimit	*integer*	Hop limit.
ifaFlags	*integer*	IFA flags.
ifaVersion	*integer*	IFA Version.
maxLength	*integer*	Max length.
opaque4	*integer*	Opaque object.
requestVector	*integer*	Request vector.

2.1432.1.2 EgrVlanXlate2IfaMdLookup1Entry

deviceId	*integer*	Device Id.
dgpp	*integer*	DGPP.
egressPortSpeed	*integer*	Egress Port Speed.
lns	*integer*	Local Namespace.
opaqueData	*integer*	Opaque object data.

2.1432.1.3 EgrVlanXlate2L2FieldsEntry

dataType **	*integer*	Data Type Valid Value : 22, 27. **Since: 4.23.0
dmac	*string*	New destination MAC address.
dvp	*integer*	Destination Virtual Port.
opaqueId	*integer*	Opaque object ID.

smac	*string*	New source MAC address.
tapp	*integer*	Index into Egress Vlan Tag Action Profile Table.
vlan	*integer*	New VLAN ID.

2.1432.1.4 EgrVlanXlate2RawEntry

entryValue	*string*	Raw dump of the entry.

2.1433 show platform trident l2 shadow ing-tag-action-profile

ingVlanTagActionProfile-Table	*table(numeric string) = IngVlanTagActionProfileEntry* see 2.1433.1	Ingress Vlan Tag Action Profile Table.

2.1433.1 IngVlanTagActionProfileEntry

dtOtagAction	*integer*	Outer VID action for double-taggged packets.
dtPotagAction	*integer*	Outer VID action for outer priority tag of double-tagged packets.
sitOcfiAction	*integer*	Inner 802.1P/PCP action for single inner-tagged packets.
sitOpriAction	*integer*	Inner CFI/DE action for single inner-tagged packets.
sitOtagAction	*integer*	Outer VID action for single inner-tagged packets.
sotOtagAction	*integer*	Outer VID Action for single outer-tagged packets.
sotPotagAction	*integer*	Outer VID action for priority outer tagged packets.
utOcfiAction	*integer*	Outer CFI/DE action for untagged packets.
utOpriAction	*integer*	Outer 802.1p/PCP action for untagged packets.
utOtagAction	*integer*	Outer VID action for untagged packets.

2.1434 show platform trident l2 shadow source-vp

vxlanSvpTable	*table(numeric string) = VxlanSourceVpEntry* see 2.1434.1	Vxlan Source VP Table.

2.1434.1 VxlanSourceVpEntry

classId	*integer*	Class ID.
cmlFlagForMoves	*integer*	CPU Managed Learning Flags for Moves.
cmlFlagForNew	*integer*	CPU Managed Learning Flags for New Entries.
disableVlanChecks	*boolean*	Disable Vlan Membership and STP checks.
disableVpPruning **	*boolean*	Disable VP Pruning. **Since: 4.23.1
dot1pProfile	*integer*	Profile Number in Ingress PRI CNG Map Table.
dot1pRemapProfile	*integer*	Profile Number in Dot1p Ingress Outer Dot1p Mapping Table.
dscpProfile	*integer*	DSCP Profile Number in DSCP_TABLE.
entryType	*integer*	Entry Type.
flexCounterBaseCounterIndex	*integer*	Base Counter Index used to compute the final counter index.
flexCounterOffsetMode	*integer*	Mode to decide the final counter index.
flexCounterPoolNumber	*integer*	Counter Pool that will be updated.
ipv4Enabled	*boolean*	IPv4 enabled.
ipv6Enabled	*boolean*	IPv6 enabled.
networkGroup **	*integer*	VP Network Group. **Since: 4.27.2
networkPort	*boolean*	Network Facing Port.
sdTagMode	*integer*	SD Tag Mode.
srcModId	*integer*	Source Module Id.
srcPort	*integer*	Source Port.
trustDscpV4	*boolean*	trust DSCP v4.
trustDscpV6	*boolean*	trust DSCP v6.
vfi	*integer*	VFI Value.

2.1435 show platform trident l2 shadow vfi

vxlanVfiTable	*table(numeric string) = VfiEntry* see 2.1435.1	Vxlan VFI Table.

2.1435.1 VfiEntry

bcIndex	*integer*	Default Multicast Group for broadcasting to this VFI.
flexCounterBaseCounterIndex	*integer*	Base Counter Index used to compute the final counter index.

flexCounterOffsetMode	*integer*	Flex Mode to decide the final counter index.
flexCounterPoolNumber	*integer*	Flex Counter Pool that will be updated.
l3Iif	*integer*	l3Iif field in the entry.
l3IifValid	*boolean*	if True, l3Iif field is valid.
protocolPktIndex	*integer*	Source Port.
stg	*integer*	Spanning Tree GroupId field in the entry.
stgValid	*boolean*	if True, stg field is valid.
umcIndex	*integer*	Default Unknown Multicast Group for this VFI.
uucIndex	*integer*	Default Unknown Unicast Group for this VFI.
vfiClassId	*integer*	vfiClassId field in the entry.
vfiClassIdValid	*boolean*	if True, vifClassId field is valid.

2.1436 show platform trident l2 shadow vlan-xlate [...]

vlanXlateTables	*table(string) = VlanXlateHwTable* see 2.1436.1	Vlan Xlate Tables for all Line Cards in a MultiChip System with the Line Card Name as key.

2.1436.1 VlanXlateHwTable

dipEntries	*table(numeric string) = VlanXlateVxlanDipEntry* see 2.1436.1.4	Destionation IP (DIP) Entries with the key being EntryId which is a hardware Index.
ovidEntries	*table(numeric string) = VlanXlateOvidEntry* see 2.1436.1.1	Outer VLAN ID based Entries with the key being EntryId which is a hardware Index.
rawEntries	*table(numeric string) = VlanXlateRawEntry* see 2.1436.1.2	All Entries in Raw Format with the key being EntryId which is a hardware index. This format is to support the raw option in CLI.
vniVfiMapEntries **	*table(numeric string) = VlanXlateVniVfiMapEntry* see 2.1436.1.3	VNI-VFI maps with the key being EntryId which is a hardware Index. **Since: 4.23.0

2.1436.1.1 VlanXlateOvidEntry

disableVlanChecks	*boolean*	Disable Vlan Membership and STP checks, if Set.
moduleId	*integer*	Module Id.
newOvid	*integer*	New Outer VLAN ID.
ovid	*integer*	Outer VLAN ID.
portNum	*integer*	Port Number.
sourceType	*integer*	Source Type : 0 for SVP and 1 for Source Global Logical Ports.
sourceVp	*integer*	Source Virtual Port Value.
tagActionProfilePtr	*integer*	Index into Ingress VLAN Tag Action Profile Table.
vlanActionValid	*boolean*	Vlan Translation Action is Valid, if Set.

2.1436.1.2 VlanXlateRawEntry

entryValue	*string*	Raw dump of the entry.

2.1436.1.3 VlanXlateVniVfiMapEntry

ovid	*integer*	Outer VLAN ID field in the entry.
tagActionProfilePtr	*integer*	Index into Ingress VLAN Tag Action Profile Table.
vfi	*integer*	VFI.
vni	*integer*	VNI.

2.1436.1.4 VlanXlateVxlanDipEntry

dip	*IP address*	DIP Prefix.

2.1437 show platform trident l2 shadow vlan-xlate-2 [...]

vlanXlate2Tables	*table(string) = VlanXlate2HwTable* see 2.1437.1	Vlan Xlate 2 Tables for all Line Cards in a MultiChip System with the Line Card Name as key.

2.1437.1 VlanXlate2HwTable

intMshipChkDisableEntries	*table(numeric string) = VlanXlate2IntMshipChkDisableEntry* see 2.1437.1.1	L2 Fields keyed by their hardware index.
rawEntries	*table(numeric string) = VlanXlate2RawEntry* see 2.1437.1.2	All Entries in Raw Format with the key being EntryId which is a hardware index. This format is to support the raw option in CLI.

2.1437.1.1 VlanXlate2IntMshipChkDisableEntry

dataLower	*integer*	Data lower.
dataUpper	*integer*	Data upper.
key	*integer*	Key.
moduleId	*integer*	Module ID.

2.1437.1.2 VlanXlate2RawEntry

entryValue	*string*	Raw dump of the entry.

2.1438 show platform trident l2 software egr-vlan-xlate-2

chipModel	*list of L2SwEgrVlanXlate2Chip* see 2.1438.1	List of flows registered in the software.

2.1438.1 L2SwEgrVlanXlate2Chip

chipName	*string*	Chip name.
flow	*list of L2SwEgrVlanXlate2Flow* see 2.1438.1.1	Flow entry registered in the software.

2.1438.1.1 L2SwEgrVlanXlate2Flow

dstMac	*string*	New destination MAC address.
dstVirtualPort	*string*	Destination virtual port.
entryId	*integer*	Flow entry Id.

opaqueId	*integer*	Opaque object ID.
srcMac	*string*	New source MAC address.
tapProfile	*integer*	Index into egress VLAN tag action profile table.
vlanId	*string*	New VLAN ID.

2.1439 show platform trident l2 software ingress-vlan-range

intfVlanRanges	*table(Interface name) = VlanRangeEntry* see 2.1439.1	Ingress VLAN ranges list.

2.1439.1 VlanRangeEntry

entryId	*integer*	EntrId of Ingress Vlan Range table.
[lag]	*interface name*	LAG Interface the interface belongs.
logicalPort	*integer*	Switchcard Logical Port number.
modId	*integer*	Switchcard module ID.
overflow	*boolean*	Inticates if the VLAN Ranges have overflown.
[vlanRanges]	*list of VlanRange* see 2.1439.1.1	List of Ingress VLAN Range to match for VLAN mapping.

2.1439.1.1 VlanRange

vlanMax	*integer*	Max VLAN ID of VLAN Range to match.
vlanMin	*integer*	Min VLAN ID of VLAN Range to match.

2.1440 show platform trident l2 software mac-address-table

macTableEntries	*list of L2SwMacAddrTableEntry* see 2.1440.1	List of MAC table entries.
numEntries	*integer*	Number of shadow entries.

2.1440.1 L2SwMacAddrTableEntry

bitmap	*integer*	MAC address bitmap.
[interface]	*interface name*	Interface for this MAC address.
keyType	*enum: vfi | int | bridge*	Mac address key type.
macAddress	*MAC address*	MAC address.

macAddressType	*enum: peerDynamicMac* | *peerSecureMac* | *pendingSecureMac* | *learnedDynamicMac* | *configuredRemoteMac* | *evpnConfiguredRemoteMac* | *dropDst* | *peerEvpnRemoteMac* | *authenticatedMac* | *peerLearnedRemoteMac* | *peerReceivedRemoteMac* | *peerAuthenticatedMac* | *learnedSecureMac* | *peerStaticMac* | *dropSrc* | *evpnIntfStaticMac* | *receivedRemoteMac* | *learnedRemoteMac* | *configuredStaticMac* | *drop* | *vplsDynamicRemoteMac* | *peerConfiguredRemoteMac* | *evpnDynamicRemoteMac* | *configuredRouterMac* | *configuredSecureMac* | *peerRouterMac* | *configuredDynamicMac* | *evpnIntfDynamicMac*	Type of the MAC address.
markedForDeletion	*boolean*	MAC address is marked for deletion.
[mcastId]	*integer*	Multicast group ID.
pending	*boolean*	MAC addres is pending.
source	*integer*	MAC address learning bitmap.
stale	*boolean*	MAC address is stale.
[virtualPort]	*integer*	Virtual port number.
[virtualPortType]	*enum: native* | *numeric* | *tagged*	Type of virtual port.
vlan	*integer*	VLAN ID.

2.1441 show platform trident l2 software mac-address-table count

numberOfShadowEntries	*integer*	Number of shadow entries.

2.1442 show platform trident l2 software mac-address-table remote count

numberOfRemoteMacs	*integer*	Number of remote MAC addresses.

2.1443 show platform trident l2 software multicast-group-table

groupCount	*L2SwMcastGroupTable-Count* see 2.1443.2	Multicast Group Table Count.
mcastGroupIds	*table(numeric string) = L2SwMcastGroupEntry* see 2.1443.1	A mapping of Mcast Group Ids to their corresponding interfaces.
[reservedMcastGroupIds]	*table(numeric string) = L2SwMcastGroupEntry* see 2.1443.1	A mapping of Reserved Mcast Group Ids to their corresponding interfaces.

2.1443.1 L2SwMcastGroupEntry

interfaces	*list of Interface name*	List of interfaces in Mcast Group.

2.1443.2 L2SwMcastGroupTableCount

numberOfGroups	*integer*	Number of groups.

2.1444 show platform trident l2 software multicast-group-table count

see 2.1443.2

2.1445 show platform trident l2 software vlan-xlate-2

chipModel	*list of L2SwVlanXlate2Chip* see 2.1445.1	List of INT Membership Check Disable Entries.

2.1445.1 L2SwVlanXlate2Chip

chipName	*string*	Chip name.
intEntry	*list of L2SwVlanXlate2Int-MshipChkDisableEntry* see 2.1445.1.1	Entries registered in the software.

2.1445.1.1 L2SwVlanXlate2IntMshipChkDisableEntry

dataLower	*integer*	Data Lower.
dataUpper	*integer*	Data Upper.
entryId	*integer*	Entry ID.
key	*integer*	Key.
moduleId	*integer*	Module ID.

2.1446 show platform trident l2 status [...]

addHosts	*integer*	Number of Add Hosts.
addMcastHosts	*integer*	Number of Mcast Hosts.
agingTimeReconfigs	*integer*	Number of Aging Time Reconfigs.
agingTimerFirings	*integer*	Number of Aging Timer Firings.
busyFailures	*integer*	Number of MAC Program Busy Failures.
clearMacEntries	*integer*	Number of Clear MAC Entries.
fullFailures	*integer*	Number of MAC Program Full Failures.
inProgressOps	*integer*	In-Progress MAC Ops.
l2modInvalidIndexReads	*integer*	Number of L2mod Invalid Index Reads.
l2modReads	*integer*	Number of L2mod Reads.
l2modSpuriousReads	*integer*	Number of L2mod Spurious Reads.
[l2modTotalEvents]	*integer*	Number of L2mod Total Events.
[l2modVxlanEvents]	*integer*	Number of L2mod Total VXLAN Events.
macProgrammerFapIds	*table(numeric string) = L2StatusMacProgrammer-Status* see 2.1446.1	Mapping from FapId to MAC Programmer Counts. Populated only for detail version of command.
[macProgrammerTotals]	*L2StatusMacProgrammer-Status* see 2.1446.1	Total Counts of all FapIds.
macTable	*integer*	Entries in MAC table.
macUpdateList	*integer*	Update Lists.
mcastAgentFapIds	*table(numeric string) = L2StatusMcastStatus* see 2.1446.2	Mapping from FapId to Mcast Agent Counts.
mcastBulkReadCmds	*integer*	Number of Multicast Bulk Read Commands.
mcastClearCmds	*integer*	Number of Multicast Clear Commands.
mcastGroups	*integer*	Number of Multicast Groups.
mcastPendingGroups	*integer*	Number of Multicast Pending Groups.
mcastReadCmds	*integer*	Number of Multicast Read Commands.
mcastWriteCmds	*integer*	Number of Multicast Write Commands.
mcastWriteCmdsForReserved-Groups	*integer*	Number of Multicast Write Commands for Reserved Groups.
otherFailures	*integer*	Number of Other MAC Program Failures.
overflowOps	*integer*	Overflow Address Ops.
parityFailures	*integer*	Number of MAC Program Parity Errors.
pendingOps	*integer*	Pending MAC Ops.
removeHosts	*integer*	Number of Remove Hosts.
syncMacEntries	*integer*	Number of Sync MAC Entries.

2.1446.1 L2StatusMacProgrammerStatus

commandQueue	*integer*	Command Queue.
commandRingConfigQueue	*integer*	Command Ring Config Queue.
commandRingStatusQueue	*integer*	Command Ring Status Queue.
l2modAgeEvents	*integer*	L2mod Aging Events.
l2modDiscardedEvents	*integer*	L2mod Discarded Events.

l2modFlushEvents	*integer*	L2mod Flush Events.
l2modLagEvents	*integer*	L2mod Lag Events.
l2modLearnEvents	*integer*	L2mod Learn Events.
l2modMulticastEvents	*integer*	L2mod Multicast Events.
l2modNonLagEvents	*integer*	L2mod Non-Lag Events.
l2modRelearnEvents	*integer*	L2mod Relearn Events.
l2modSwLagEvents	*integer*	L2mod Software Lag Events.
l2modTotalEvents	*integer*	L2mod Total Events.
l2modUnexpectedEvents	*integer*	L2mod Unexpected Events.

2.1446.2 L2StatusMcastStatus

mcastBulkReadCmds	*integer*	Number of Multicast Bulk Read Commands.
mcastClearCmds	*integer*	Number of Multicast Clear Commands.
mcastReadCmds	*integer*	Number of Multicast Read Commands.
mcastUnsyncedGroups	*integer*	Number of Multicast Unsynced Groups.
mcastWriteCmds	*integer*	Number of Multicast Write Commands.
mcastWriteCmdsForReservedGroups	*integer*	Number of Multicast Write Commands for Reserved Groups.

2.1447 show platform trident l3 cmdring

l3AgentStatus	*L3AgentStatus* see 2.1447.2	L3 agent status.
l3CmdRing	*CmdRing* see 2.1447.1	L3 schan command ring summary.

2.1447.1 CmdRing

cmdRingFull	*boolean*	True if CmdRing buffer is full.
cmdRingFullCount	*integer*	Number of times CmdRing buffer has been full.
dlbEcmpFlowset **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetMember **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe0 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe0 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe1 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe1 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe2 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe2 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe3 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe3 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe4 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe4 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe5 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe5 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe6 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe6 command statistics. **Since: 4.24.2
dlbEcmpFlowsetMemberPipe7 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset Member Pipe7 command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe0 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe1 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe2 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe3 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe4 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe5 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2

dlbEcmpFlowsetPipe6 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpFlowsetPipe7 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Flowset command statistics. **Since: 4.24.2
dlbEcmpGlbQuantizeThreshold **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Glb Quantize Threshold command statistics. **Since: 4.23.0
dlbEcmpGroupControl **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Control command statistics. **Since: 4.20.0
dlbEcmpGroupMembership **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Membership command statistics. **Since: 4.20.0
dlbEcmpGroupNhiMembers0To12 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Nhi Members 0 To 12 command statistics. **Since: 4.23.0
dlbEcmpGroupNhiMembers13-To25 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Nhi Members 13 To 25 command statistics. **Since: 4.23.0
dlbEcmpGroupNhiMembers26-To38 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Nhi Members 26 To 38 command statistics. **Since: 4.23.0
dlbEcmpGroupNhiMembers39-To51 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Nhi Members 39 To 51 command statistics. **Since: 4.23.0
dlbEcmpGroupNhiMembers52-To63 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Nhi Members 52 To 63 command statistics. **Since: 4.23.0
dlbEcmpGroupPortMembers0-To15 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Port Members 0 To 15 command statistics. **Since: 4.23.0
dlbEcmpGroupPortMembers16-To31 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Port Members 16 To 31 command statistics. **Since: 4.23.0
dlbEcmpGroupPortMembers32-To47 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Port Members 32 To 47 command statistics. **Since: 4.23.0
dlbEcmpGroupPortMembers48-To63 **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Port Members 48 To 63 command statistics. **Since: 4.23.0
dlbEcmpGroupPortToMember **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Group Port To Member command statistics. **Since: 4.20.0
dlbEcmpPortQualityMapping **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Port Quality Mapping command statistics. **Since: 4.23.0
dlbEcmpPortQualityUpdate-MeasureControl **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Port Quality Update Measure Cotrol command statistics. **Since: 4.24.0
dlbEcmpQuantizeControl **	*SchanCmd* see 2.1447.1.1	Dlb Ecmp Quantize Control command statistics. **Since: 4.23.0

ecmp	*SchanCmd* see 2.1447.1.1	Ecmp command statistics.
ecmpDlbControl **	*SchanCmd* see 2.1447.1.1	Ecmp Dlb Control command statistics. **Since: 4.20.0
ecmpGroup	*SchanCmd* see 2.1447.1.1	EcmpGroup command statistics.
egrIpTunnel	*SchanCmd* see 2.1447.1.1	EgrIpTunnel command statistics.
egrIpTunnelIpv6	*SchanCmd* see 2.1447.1.1	EgrIpTunnelIpv6 command statistics.
egrMacDaProfile	*SchanCmd* see 2.1447.1.1	EgrMacDaProfile command statistics.
egrMplsVcAndSwapLabel	*SchanCmd* see 2.1447.1.1	EgrMplsVcAndSwapLabel command statistics.
ignoreTotal	*integer*	Total number of ignored schan commands.
ingActiveL3IifProfile	*SchanCmd* see 2.1447.1.1	IngActiveL3IifProfile command statistics.
ingDvpTable	*SchanCmd* see 2.1447.1.1	IngDvpTable command statistics.
interface	*SchanCmd* see 2.1447.1.1	Interface command statistics.
ipMulticastTcam	*SchanCmd* see 2.1447.1.1	IpMulticastTcam command statistics.
ipOptionControlProfileTable **	*SchanCmd* see 2.1447.1.1	Ip Option Control Profile Table command statistics. **Since: 4.30.0
l2Mc	*SchanCmd* see 2.1447.1.1	L2Mc command statistics.
l3AlpmIp4Ucast	*SchanCmd* see 2.1447.1.1	L3AlpmIp4Ucast command statistics.
l3AlpmIp6Ucast	*SchanCmd* see 2.1447.1.1	L3AlpmIp6Ucast command statistics.
l3DefIp4Ucast	*SchanCmd* see 2.1447.1.1	L3DefIp4Ucast command statistics.
l3DefIp6Ucast	*SchanCmd* see 2.1447.1.1	L3DefIp6Ucast command statistics.
l3DefipAlpmLevel2Ecc **	*SchanCmd* see 2.1447.1.1	L3DefipAlpmLevel2Ecc command statistics. **Since: 4.21.3
l3DefipAlpmLevel3Ecc **	*SchanCmd* see 2.1447.1.1	L3DefipAlpmLevel3Ecc command statistics. **Since: 4.21.3
l3EcmpGroupHierarchical **	*SchanCmd* see 2.1447.1.1	EcmpGroupHierarchical command statistics. **Since: 4.27.2
l3Entry4Mcast	*SchanCmd* see 2.1447.1.1	L3Entry4Mcast command statistics.
l3Entry4Ucast	*SchanCmd* see 2.1447.1.1	L3Entry4Ucast command statistics.
l3Entry6Mcast **	*SchanCmd* see 2.1447.1.1	L3Entry6Mcast command statistics. **Since: 4.22.0

l3Entry6Ucast	*SchanCmd* see 2.1447.1.1	L3Entry6Ucast command statistics.
l3EntryEcc	*SchanCmd* see 2.1447.1.1	L3EntryEcc command statistics.
l3EntryOnlyEcc **	*SchanCmd* see 2.1447.1.1	L3EntryOnlyEcc command statistics. **Since: 4.20.5
l3IifProfile	*SchanCmd* see 2.1447.1.1	L3IifProfile command statistics.
l3Ipmc	*SchanCmd* see 2.1447.1.1	L3Ipmc command statistics.
l3Ipmc1	*SchanCmd* see 2.1447.1.1	L3Ipmc1 command statistics.
l3IpmcRemap	*SchanCmd* see 2.1447.1.1	L3IpmcRemap command statistics.
l3Tunnel	*SchanCmd* see 2.1447.1.1	L3Tunnel command statistics.
l3TunnelEcc **	*SchanCmd* see 2.1447.1.1	L3TunnelEcc command statistics. **Since: 4.23.1
maxCmdRingSize	*integer*	Maximum size in the history of CmdRing buffer.
mmuReplCopyCount	*SchanCmd* see 2.1447.1.1	MmuReplCopyCount command statistics.
mmuReplHead	*SchanCmd* see 2.1447.1.1	MmuReplHead command statistics.
mmuReplInfo0	*SchanCmd* see 2.1447.1.1	MmuReplInfo0 command statistics.
mmuReplInfo1	*SchanCmd* see 2.1447.1.1	MmuReplInfo1 command statistics.
mmuReplList	*SchanCmd* see 2.1447.1.1	MmuReplList command statistics.
mplsEntry	*SchanCmd* see 2.1447.1.1	MplsEntry command statistics.
mplsEntryEcc **	*SchanCmd* see 2.1447.1.1	MplsEntryEcc command statistics. **Since: 4.20.0
mplsNextHop	*SchanCmd* see 2.1447.1.1	MplsNextHop command statistics.
mtuAttribute	*SchanCmd* see 2.1447.1.1	MtuAttribute command statistics.
mtuAttributeIndex	*SchanCmd* see 2.1447.1.1	MtuAttributeIndex command statistics.
mtuValues	*SchanCmd* see 2.1447.1.1	MtuValues command statistics.
myStationTcam	*SchanCmd* see 2.1447.1.1	MyStationTcam command statistics.
name	*string*	Name to give to the summary of all schan commands.
nextHop	*SchanCmd* see 2.1447.1.1	NextHop command statistics.
retryTotal	*integer*	Total number of retried schan commands.
sdTagNextHop	*SchanCmd* see 2.1447.1.1	SdTagNextHop command statistics.

successTotal	*integer*	Total number of successful schan commands.
vxlanDip6MplsEntry **	*SchanCmd* see 2.1447.1.1	VxlanDip6MplsEntry command statistics. **Since: 4.23.2
vxlanEgrDvpAttribute	*SchanCmd* see 2.1447.1.1	VxlanEgrDvpAttribute command statistics.
vxlanSip6MplsEntry **	*SchanCmd* see 2.1447.1.1	VxlanSip6MplsEntry command statistics. **Since: 4.23.2
vxlanSipMplsEntry	*SchanCmd* see 2.1447.1.1	VxlanSipMplsEntry command statistics.
vxlanVniMplsEntry	*SchanCmd* see 2.1447.1.1	VxlanVniMplsEntry command statistics.

2.1447.1.1 SchanCmd

ignoreCnt	*integer*	Schan command ignore count.
retryCnt	*integer*	Schan command retry count.
successCnt	*integer*	Schan command success count.

2.1447.2 L3AgentStatus

bulkDownloads	*table(numeric string) = integer*	Bulk download statistics. Key is module ID, value is number of bulk downloads.
coprocessors	*integer*	Number of coprocessors.
coprocessorsAttached	*integer*	Number of attached coprocessors.
hwParityErrors	*table(string) = integer*	Number of hardware parity errors. Key is hardware table id, value is number of parity errors.
pendingCmds	*integer*	Number of pending schan commands.
totalCmds	*integer*	Total number of schan commands.
ucastAdjsV4	*integer*	Total number of IPv4 unicast adjacencies.
ucastAdjsV6	*integer*	Total number of IPv6 unicast adjacencies.
ucastPendingAdjsV4	*integer*	Number of IPv4 unicast adjacencies pending hardware programming.
ucastPendingAdjsV6	*integer*	Number of IPv6 unicast adjacencies pending hardware programming.
ucastPendingRoutesV4	*integer*	Number of IPv4 unicast routes pending hardware programming.
ucastPendingRoutesV6	*integer*	Number of IPv6 unicast routes pending hardware programming.
ucastRoutesV4	*integer*	Total number of IPv4 unicast routes.
ucastRoutesV6	*integer*	Total number of IPv6 unicast routes.
ucastUnprogrammedAdjsV4	*integer*	Number of IPv4 unicast adjacencies that failed hardware programming.
ucastUnprogrammedAdjsV6	*integer*	Number of IPv6 unicast adjacencies that failed hardware programming.
ucastUnprogrammedRoutesV4	*integer*	Number of IPv4 unicast routes that failed hardware programming.
ucastUnprogrammedRoutesV6	*integer*	Number of IPv6 unicast routes that failed hardware programming.

2.1448 show platform trident l3 shadow (my-station | my-station-2) [...]

l3MyStation	*table(numeric string) = L3MyStationEntry* see 2.1448.1	Collection of MyStation objects keyed by the entry ID.

2.1448.1 L3MyStationEntry

arp	*boolean*	Arp termination.
cpu	*boolean*	Copy to CPU enabled.
dscd	*boolean*	Discard enabled.
ingressPort	*integer*	Ingress port for this entry.
ingressPortMask	*integer*	Mask for this ingress port.
macAddress	*MAC address*	MAC address of this entry.
macMask	*MAC address*	Mask for the MAC address of the entry.
modId	*integer*	The module ID of the entry.
modIdMask	*integer*	The moduleId mask of the entry.
mpls	*boolean*	MPLS termination.
[profileId] **	*integer*	Profile Id. **Since: 4.21.3
trunkBit	*integer*	Trunk bit.
trunkBitMask	*integer*	Trunk bit mask.
[v4Multicast]	*boolean*	IPv4 Multicast routing enabled.
v4Unicast	*boolean*	IPv4 Unicast routing enabled.
[v6Multicast]	*boolean*	IPv6 Multicast routing enabled.
v6Unicast	*boolean*	IPv6 Unicast routing enabled.
vlan	*integer*	The VLAN the entry belongs to.
vlanMask	*integer*	VLAN mask for this entry.

2.1449 show platform trident l3 shadow counter [...]

revision 2

see 2.1428

2.1450 show platform trident l3 shadow dlb-ecmp-group-control [...]

dlbEcmpGroupControl	*table(numeric string) = L3ShwDlbEcmpGroup-ControlEntry* see 2.1450.1	Individual DLB_ECMP_GROUP_CONTROL keyed by entryId.

2.1450.1 L3ShwDlbEcmpGroupControlEntry

baseAddr	*integer*	Group Port To Member Base Address.
flowBase	*integer*	Flow Set Base.
flowSize	*integer*	Flow Set Size.
groupSize	*integer*	Number of members in Primary Group.
inDur	*integer*	Inactivity Duration.

memPtr	*integer*	Group Membership Pointer.
mode	*integer*	Port Assignment Mode.
oc	*integer*	Optimal Candidate.
path	*integer*	Primary Path Threshold.

2.1451 show platform trident l3 shadow dlb-ecmp-group-membership [...]

dlbEcmpGroupMembership	*table(numeric string) = L3ShwDlbEcmpGroupMembershipEntry* see 2.1451.1	Individual DLB_ECMP_GROUP_MEMBERSHIP keyed by DLB ID.

2.1451.1 L3ShwDlbEcmpGroupMembershipEntry

portMap	*string*	Indicates whether a logical port is a member of a DLB group.

2.1452 show platform trident l3 shadow dlb-ecmp-group-port-to-member [...]

dlbEcmpGroupPortToMember	*table(numeric string) = L3ShwDlbEcmpGroupPortToMemberEntry* see 2.1452.1	Individual DLB_ECMP_GROUP_PORT_TO_MEMBER keyed by MaxPort * DLB id + port number.

2.1452.1 L3ShwDlbEcmpGroupPortToMemberEntry

ecmpBitmap	*string*	Indicate if this port is part of DLB.

2.1453 show platform trident l3 shadow ecmp-dlb-control [...]

ecmpDlbControl	*table(numeric string) = L3ShwEcmpDlbControlEntry* see 2.1453.1	Individual ECMP_DLB_CONTROL keyed by ECMP group entryId.

2.1453.1 L3ShwEcmpDlbControlEntry

dlbId	*integer*	DLB Id of the ECMP group.
enable	*boolean*	DLB group is enabled.

2.1454 show platform trident l3 shadow interfaces [...]

l3Intf	*table(numeric string) = L3ShwIntfEntry* see 2.1454.1	Individual L3 Interfaces keyed by interface value.

2.1454.1 L3ShwIntfEntry

group	*integer*	The multicast group the interface is in.
innerVlan	*integer*	The inner vlan.
ipmcIif	*integer*	IP Mutlicast IIF.
sourceMac	*MAC address*	The MAC address of this interface.
ttl	*integer*	The TTL value of the interface.
tunnel	*integer*	The tunnel index.
unit	*integer*	The forwarding chip the interface belongs to.
vlan	*integer*	The vlan for the interface.
vrf	*integer*	The VRF the interface belongs to.

2.1455 show platform trident l3 shadow l3-iif-profile [...]

l3IifProfile	*table(numeric string) = L3ShwIifProfileEntry* see 2.1455.1	Individual L3 iif profile entries indexed by entry ID.

2.1455.1 L3ShwIifProfileEntry

allowGlobalRouting	*boolean*	Allow global routing.
icmpToCpu	*boolean*	ICMP messages to CPU enabled.
trustDscpPtr	*integer*	Pointer to trust DSCP.
uRPFDefRc	*boolean*	URPF Default route check enabled.
uRPFMode	*integer*	The URPF mode of the interface.
unit	*integer*	The forwarding chip the l3-iif-profile belongs to.
unkMC2CpuV4	*boolean*	v4 unknown multicast to CPU.
unkMC2CpuV6	*boolean*	v6 unknown multicast to CPU.
v4Multicast	*boolean*	v4 multicast routing enabled.
v4Unicast	*boolean*	v4 unicast routing enabled.
v6Multicast	*boolean*	v6 multicast routing enabled.
v6Unicast	*boolean*	v6 unicast routing enabled.

2.1456 show platform trident l3 shadow l3-tunnel [...]

revision 2

l3Tunnel	*table(numeric string) = L3ShwL3TunnelEntry* see 2.1456.1	Individual L3 tunnel keyed by entryId.

2.1456.1 L3ShwL3TunnelEntry

Iif	*integer*	L3 IIF index entry points to.
TT	*integer*	Tunnel Type.
dip	*IP address*	Destination IP address.
endId	*integer*	End Index of entry.

grePayload4	*integer*	GRE Payload Ipv4.
grePayload6	*integer*	GRE Payload Ipv6.
payload4	*integer*	Payload Ipv4.
payload6	*integer*	Payload Ipv6.
proto	*integer*	Protocol.
sip **	*IP address*	Source IP address. **Since: 4.22.1
startId	*integer*	Start Index of entry.
subTT	*integer*	SubTunnel Type.
unit	*integer*	The forwarding chip l3_tunnel belongs to.

2.1457 show platform trident l3 shadow mroutes [...]

mRoutes	*table(numeric string) = L3ShwMroutesEntry* see 2.1457.1	Multicast Routing table, indexed by the entry ID.

2.1457.1 L3ShwMroutesEntry

classId	*integer*	Class ID.
fDrop	*string*	Fast Drop.
groupId	*integer*	Group ID.
grpAddr	*IP address*	Multicast groupaddress.
[rpaId]	*integer*	RPA ID.
srcAddr	*IP address*	Multicast source address.
vlan	*integer*	VLAN ID for the entry.
vrf	*integer*	VRF.

2.1458 show platform trident l3 shadow mtu [...]

l3Mtu	*table(numeric string) = L3ShwMtuEntry* see 2.1458.1	Individual L3 MTUs keyed by interface value.

2.1458.1 L3ShwMtuEntry

mtu	*integer*	Maximum transmission unit for this interface.

2.1459 show platform trident l3 shadow next-hops her

l3NextHopsHer	*table(numeric string) = L3ShwNextHopsHerEntry* see 2.1459.1	HER next hops table, indexed by the entryId.

2.1459.1 L3ShwNextHopsHerEntry

drop	*boolean*	Drop enabled.
dvp	*integer*	DVP for the entry.
intf	*integer*	Interface for the entry.
mac	*MAC address*	MAC address of the entry.
macDaIdx	*integer*	MAC DA Index for the entry.
modId	*integer*	Module ID for the entry.
port	*integer*	Port for the entry.
toCpu	*boolean*	Send to CPU enabled.
vlan	*integer*	VLAN ID for the entry.

2.1460 show platform trident l3 shadow next-hops multipath [...]

l3NextHopsEcmp	*table(numeric string) = L3NextHopsEcmp* see 2.1460.1	ECMP next hops table, indexed by the entry ID.

2.1460.1 L3NextHopsEcmp

intfs	*list of integer*	The interfaces associated with the multipath next hop entry ID.

2.1461 show platform trident l3 shadow next-hops multipath hfec [...]

l3HierarchicalEcmpGroup	*table(numeric string) = L3NextHopsEcmpHierarchicalList* see 2.1461.1	Hierarchical ECMP group table, indexed by the entry ID.

2.1461.1 L3NextHopsEcmpHierarchicalList

members	*list of L3NextHopsEcmpHierarchicalMember* see 2.1461.1.1	The members associated with the multipath next hop entry ID.

2.1461.1.1 L3NextHopsEcmpHierarchicalMember

overlayNextHop	*integer*	Overlay next hop index.
underlayEcmp	*boolean*	Underlay next hop is ECMP.
underlayNextHop	*integer*	Underlay next hop index.

2.1462 show platform trident l3 shadow next-hops simple [...]

l3NextHopsSimple	*table(numeric string) = L3ShwNextHopsSimpleEntry* see 2.1462.1	Simple next hops table, indexed by the entryId.

2.1462.1 L3ShwNextHopsSimpleEntry

drop	*boolean*	Drop enabled.
intf	*integer*	Interface for the entry.
mac	*MAC address*	MAC address of the entry.
macDaIdx	*integer*	Mac DA Index for the entry.
modId	*integer*	Module ID for the entry.
mplsLabel	*integer*	MPLS Label for the entry.
port	*integer*	Port for the entry.
toCpu	*boolean*	Send to CPU enabled.
vlan	*integer*	VLAN ID for the entry.

2.1463 show platform trident l3 shadow routes defip [...]

freeV6Entries	*integer*	Free IPv6 entries available.
ipVersion	*enum: v6 | v4*	The IP version of table in this object, either v4 or v6.
l3Routes	*table(string) = L3ShwRoutesDefipEntry* see 2.1463.1	Defip routing table, indexed by the entry ID.
maxEcmpPaths	*integer*	Max number of ECMP paths in v6.
totalDefIpEntries	*integer*	The number of DEFIP entries in Unit 0.

2.1463.1 L3ShwRoutesDefipEntry

bktId	*integer*	Bucket ID.
classId	*integer*	Class ID.
ecmp	*boolean*	ECMP Nexthop.
hit	*string*	Hit counter.
hitIdx	*integer*	Hit Index.
index	*integer*	LPM index.
intf	*integer*	Interface.
macAddress	*MAC address*	Next hop MAC address.
modId	*integer*	Module ID.
netAddr	*IP address prefix*	IP address.
port	*integer*	Port.
priority	*integer*	Priority.
vlan	*string*	Vlan.
vrf	*integer*	VRF.

2.1464 show platform trident l3 shadow routes host [...]

freeTableEntries	*integer*	The number of free table entries, L3 table for IP v4, IPv6 table for IP v6.
ipVersion	*enum: v6 | v4*	The IP version of table in this object, either v4 or v6.
l3Routes	*table(numeric string) = L3ShwRoutesHostEntry* see 2.1464.1	Host Routing table, indexed by the entryId.

2.1464.1 L3ShwRoutesHostEntry

classId	*integer*	Class ID.
ecmp	*boolean*	ECMP Nexthop.
hit	*string*	Hit counter.
intf	*integer*	L3 interface.
ipAddr	*IP address*	IP address.
macAddress	*MAC address*	The MAC address.
modId	*integer*	Module ID.
port	*integer*	Port.

`vrf`	*integer*	VRF.

2.1465 show platform trident l3 shadow routes lpm [...]

freeV6Entries	*integer*	Free IPv6 entries available.
ipVersion	*enum: v6 | v4*	The IP version of table in this object, either v4 or v6.
l3Routes	*table(string) = L3ShwRoutesLpmEntry* see 2.1465.1	LPM routing table, indexed by the entry ID.
maxEcmpPaths	*integer*	Max number of ECMP paths in v6.
totalDefIpEntries	*integer*	The number of DEFIP entries in Unit 0.

2.1465.1 L3ShwRoutesLpmEntry

classId	*integer*	Class ID.
ecmp	*boolean*	ECMP Nexthop.
hit	*string*	Hit counter.
index	*integer*	LPM index.
intf	*integer*	Interface.
macAddress	*MAC address*	Next hop MAC address.
modId	*integer*	Module ID.
netAddr	*IP address prefix*	IP address.
port	*integer*	Port.
priority	*integer*	Priority.
vlan	*string*	Vlan.
vrf	*integer*	VRF.

2.1466 show platform trident l3 shadow routes-v6 defip [...]

see 2.1463

2.1467 show platform trident l3 shadow routes-v6 host [...]

see 2.1464

2.1468 show platform trident l3 shadow routes-v6 lpm [...]

see 2.1465

2.1469 show platform trident l3 shadow tunnel TYPE [...]

l3EgrIpTunnel	*table(numeric string) = L3ShwEgrIpTunnelEntry* see 2.1469.1	Individual L3 egr ip tunnels keyed by entryId.

2.1469.1 L3ShwEgrIpTunnelEntry

dscpSel	*integer*	The DSCP select value of the tunnel.
dstIp	*IP address*	The destination IP address of the tunnel.
entryType	*string*	The type of entry.
srcIp	*IP address*	The source IP address of the tunnel.
ttl	*integer*	The TTL of the IP tunnel.
tunnelType	*integer*	The type of tunnel.
unit	*integer*	The forwarding chip the IP tunnel belongs to.

2.1470 show platform trident l3 shadow unified-tcam

unifiedTcamEntries	*table(numeric string) = UnifiedTcamEntry* see 2.1470.1	Collection of UnifiedTcamEntry objects keyed by entry ID.

2.1470.1 UnifiedTcamEntry

bankId	*integer*	Unified TCAM Bank ID.

2.1471 show platform trident l3 shadow unified-tcam map

unifiedTcamMapEntries	*table(numeric string) = UnifiedTcamMapEntry* see 2.1471.1	Collection of UnifiedTcamMapEntry objects keyed by entry ID.

2.1471.1 UnifiedTcamMapEntry

banks	*list of UnifiedTcamBankId* see 2.1471.1.1	Unified TCAM Bank Map.

2.1471.1.1 UnifiedTcamBankId

bankId	*integer*	Unified TCAM Bank Id.

2.1472 show platform trident l3 software alpm-table [...]

revision 2

[l3AlpmRouteLpm]	*L3AlpmRouteLpm* see 2.1472.2	LPM route.
[l3AlpmRoutes]	*L3AlpmRoutes* see 2.1472.3	List of IPv4 routes.
[l3AlpmRoutesV6]	*L3AlpmRoutes* see 2.1472.3	List of IPv4 routes.
[l3AlpmTableBuckets]	*L3AlpmTableBuckets* see 2.1472.4	ALPM Buckets.
[l3AlpmTableState]	*AlpmTableState* see 2.1472.1	ALPM table.
[l3AlpmThreeLevelTable-Stats]	*L3AlpmThreeLevelTable-Stats* see 2.1472.5	3 Level ALPM Statistics.
[l3AlpmTwoLevelTableStats]	*L3AlpmTwoLevelTableStats* see 2.1472.6	2 Level ALPM Statistics.
[l3AlpmUnprogL2Routes]	*L3AlpmRoutes* see 2.1472.3	List of unprogrammed IPv4 L2 pivots.
[l3AlpmUnprogL2RoutesV6]	*L3AlpmRoutes* see 2.1472.3	List of unprogrammed IPv4 L2 pivots.
[l3AlpmUnprogL3Routes]	*L3AlpmRoutes* see 2.1472.3	List of unprogrammed IPv4 L3 routes.
[l3AlpmUnprogL3RoutesV6]	*L3AlpmRoutes* see 2.1472.3	List of unprogrammed IPv4 L3 routes.

2.1472.1 AlpmTableState

alpmTableMode	*integer*	ALPM table mode.
bucketsV4	*integer*	Number of IPv4 buckets in use.
bucketsV6Full	*integer*	Number of /128 IPv6 buckets in use.
bucketsV6Half	*integer*	Number of /64 IPv6 buckets in use.
fibRoutesV4	*integer*	Number of IPv4 entries in UFT table.
fibRoutesV6Full	*integer*	Number of /128 IPv6 entries in the UFT table.
fibRoutesV6Half	*integer*	Number of /64 IPv6 entries in the UFT table.
maxAlpmTableEntries	*integer*	Maximum number of entries in ALPM table.
maxFibRoutesV4	*integer*	Maximum possible number of IPv4 entries in UFT table.
maxFibRoutesV6Full	*integer*	Maximum possible number of /128 IPv6 entries in the UFT table.
maxFibRoutesV6Half	*integer*	Maximum possible number of /64 IPv6 entries in the UFT table.
pivotsV4	*integer*	Number of IPv4 entries in DEFIP table.
pivotsV6Full	*integer*	Number of /128 IPv6 entries in DEFIP table.
pivotsV6Half	*integer*	Number of /64 IPv6 entries in DEFIP table.

2.1472.2 L3AlpmRouteLpm

[fibBank]	*integer*	Fib bank.
[fibBucket]	*integer*	Fib bucket.
[fibEntry]	*integer*	Fib entry.
[fibPos]	*integer*	Fib Position.
[fibRoute]	*IP address prefix*	Fib route.
[fibSubBucket]	*integer*	Fib sub-bucket.
[l1Pivot]	*IP address prefix*	L1 pivot.
[l1PivotBucket]	*integer*	L1 pivot bucket.
[l1PivotCoveringPrefix]	*IP address prefix*	L1 pivot covering prefix.
[l1PivotRouteCount]	*integer*	L1 pivot route count.
[pivot]	*IP address prefix*	Pivot.
[pivotBucket]	*integer*	Pivot bucket.
[pivotCoveringPrefix]	*IP address prefix*	Pivot covering prefix.
[pivotRouteCount]	*integer*	Pivot route count.
prefix	*IP address prefix*	IP Route Address Prefix.
vrf	*integer*	VRF.

2.1472.3 L3AlpmRoutes

l3AlpmRoutes	*table(numeric string) = L3AlpmRouteAttributes* see 2.1472.3.1	ALPM routes indexed by physical position in ALPM.

2.1472.3.1 L3AlpmRouteAttributes

[bank]	*integer*	Bank Id.
[bucket]	*integer*	Bucket Id.
[entry]	*integer*	Entry Id.
route	*IP address prefix*	Route prefix.
[subBucket]	*integer*	SubBucket Id.
vrf	*integer*	VRF Id.

2.1472.4 L3AlpmTableBuckets

buckets	*list of L3AlpmBucket* see 2.1472.4.1	List of buckets in use.
numBuckets	*integer*	Number of buckets.
[numBucketsPerPage]	*integer*	Number of buckets / page.
[numDefragsV4]	*integer*	Number of V4 defrags.
[numDefragsV6]	*integer*	Number of V6 defrags.
[numPages]	*integer*	Number of pages.
[routeTypes]	*table(string) = L3AlpmBucketsRouteType* see 2.1472.4.2	A mapping of pivot route types to their counters.
uRPF	*boolean*	uRPF is enabled.

2.1472.4.1 L3AlpmBucket

entry	*integer*	Id of bucket entry.
freeBitmap	*integer*	Free bit map.
page	*integer*	Pages allocated.
routeType	*enum: v6Half | v4 | v6Full*	Route type.
subBuckets	*list of L3AlpmSubBucket* see 2.1472.4.1.1	List of sub-buckets.

2.1472.4.1.1 L3AlpmSubBucket

allocated	*integer*	Number of entries allocated for pivot route.
pivot	*IP address prefix*	Pivot route identifier.
vrf	*integer*	VRF of pivot route.

2.1472.4.2 L3AlpmBucketsRouteType

bucketsPerPivot	*integer*	Number of buckets / pivot for this route type.
entriesPerPivot	*integer*	Number of entries / pivot for this route type.

2.1472.5 L3AlpmThreeLevelTableStats

batchedRequests	*integer*	Number of Batched Requests.
batches	*integer*	Batches.
batchesCleanseTime	*floating point number*	CLeanse Time for Batches.
cleanses	*integer*	Cleanses.
l1PivotsFree	*integer*	L1 Pivots Free Percentage.
l1PivotsPeak	*integer*	L1 Pivots peak.
l1PivotsTotal	*integer*	Total of L1 Pivots.
l1PivotsUsed	*integer*	L1 Pivots Used.
l2BanksBucketRows	*integer*	This represents the smem_prows.
l2BanksFormat1	*integer*	L2 Banks for Format 1.
l2BanksFormat2	*integer*	L2 Banks for Format 2.
l2BanksFormat3	*integer*	L2 Banks for Format 3.
l2BanksFormat4	*integer*	L2 Banks for Format 4.
l2BanksFormat5	*integer*	L2 Banks for Format 5.
[l2BanksFormat6] **	*integer*	L2 Banks for Format 6. **Since: 4.27.2
[l2BanksFormat7] **	*integer*	L2 Banks for Format 7. **Since: 4.27.2
[l2BanksFormat8] **	*integer*	L2 Banks for Format 8. **Since: 4.27.2
[l2BanksFormat9] **	*integer*	L2 Banks for Format 9. **Since: 4.27.2
l2BanksPeak	*integer*	L2 Banks Peak.
l2BanksPercent	*integer*	L2 Banks Percentage.
l2BanksTotalPoolsPerRows	*integer*	L2 Banks Total Pools Per Rows.
l2BanksUsed	*integer*	L2 Banks Used.

l2BucketsPeakRows	*integer*	L2 Buckets Peak Rows.
l2BucketsRowsPercent	*integer*	L2 Buckets Rows Percentage.
l2BucketsRowsUsed	*integer*	L2 Buckets Rows Used.
l2PivotsFree	*integer*	L2 Pivots Free Percentage.
l2PivotsPeak	*integer*	L2 Pivots Peak.
l2PivotsTotal	*integer*	Total of L2 Pivots.
l2PivotsUsed	*integer*	L2 Pivots Used.
l2V4PivotsUnprogrammed	*integer*	L2 V4 Pivots Unprogrammed.
l2V6PivotsUnprogrammed	*integer*	L2 V6 Pivots Unprogrammed.
l3BanksBucketRows	*integer*	This represents the smem_prows.
l3BanksFormat1	*integer*	L3 Banks for Format 1.
l3BanksFormat2	*integer*	L3 Banks for Format 2.
l3BanksFormat3	*integer*	L3 Banks for Format 3.
l3BanksFormat4	*integer*	L3 Banks for Format 4.
l3BanksFormat5	*integer*	L3 Banks for Format 5.
l3BanksFormat6	*integer*	L3 Banks for Format 6.
l3BanksFormat7	*integer*	L3 Banks for Format 7.
l3BanksFormat8	*integer*	L3 Banks for Format 8.
l3BanksFormat9	*integer*	L3 Banks for Format 9.
l3BanksPeak	*integer*	L3 Banks Peak.
l3BanksPercent	*integer*	L3 Banks Percentage.
l3BanksTotalPoolsPerRows	*integer*	L2 Banks Total Pools Per Rows.
l3BanksUsed	*integer*	L3 Banks Used.
l3BucketsPeakRows	*integer*	L3 Buckets Peak Rows.
l3BucketsRowsPercent	*integer*	L2 Buckets Rows Percentage.
l3BucketsRowsUsed	*integer*	L3 Buckets Rows Used.
numRequests	*integer*	Number of Requests.
peakCleanseTime	*floating point number*	Peak Cleanse Time.
requestHandleTime	*floating point number*	Request Handling Time.
v4ConfigRoutes	*integer*	v4 Configuration Routes.
v4StatusRoutes	*integer*	v4 Status Routes.
v4UnprogrammedRoutes	*integer*	v4 Unprogrammed Routes.
v6ConfigRoutes	*integer*	v6 Configuration Routes.
v6L1Pivots	*integer*	v6 L1 Pivots.
v6L2Pivots	*integer*	v6 L2 Pivots.
v6StatusRoutes	*integer*	v6 Status Routes.
v6UnprogrammedRoutes	*integer*	v6 Unpprogrammed Routes.

2.1472.6 L3AlpmTwoLevelTableStats

v4Route	*L3AlpmStatsRouteType* see 2.1472.6.1	v4Route.
v6FullHalfPivot	*integer*	Move Pivot v6Full to v6Half.
v6FullHalfRoute	*integer*	Move Routes v6Full to v6Half.
v6FullRoute	*L3AlpmStatsRouteType* see 2.1472.6.1	v6FullRoute.
v6HalfFullPivot	*integer*	Move Pivot v6Half to v6Full.
v6HalfFullRoute	*integer*	Move Routes v6Half to v6Full.

v6HalfRoute	*L3AlpmStatsRouteType* see 2.1472.6.1	v6HalfRoute.

2.1472.6.1 L3AlpmStatsRouteType

bucketCreateFailWithSwap	*integer*	Bucket Create Fail with Swap.
bucketCreateFailWithoutSwap	*integer*	Bucket Create Fail Without Swap.
[deleteMerges]	*integer*	Delete Merge.
[deleteMergesFail]	*integer*	Delete Merges Fail.
[deleteMergesToChild]	*integer*	Delete Merges to Child.
[deleteMergesToChildFail]	*integer*	Delete Merges to Child Fail.
[deleteMergesToChildFail-Node]	*integer*	Delete Merge to Child Fail Node.
[deleteMergesToParent]	*integer*	Delete Merges to Parent.
[deleteMergesToParentFail]	*integer*	Delete Merges to Parent Fail.
[deleteMergesToParentFail-Node]	*integer*	Delete Merges to Parent Fail Node.
[deleteMergesToParentFail-Tcam]	*integer*	Delete Merges to Parent Fail Tcam.
[deleteMergesToParentNon-Pivot]	*integer*	Delete Merges to Parent Non Pivot.
[deleteMergesToParentPivot]	*integer*	Delete Merges to Parent Pivot.
[splitMerges]	*integer*	Splits Merges.
[splitMergesFail]	*integer*	Split Merges Fail.
[splitMergesToChild-ExistingPivot]	*integer*	Split Merges Child Existing Pivot.
[splitMergesToChild-ExistingPivotFail]	*integer*	Split Merges to Child Fail.
[splitMergesToChildSplit-Pivot]	*integer*	Split Merges to Child Split Pivot.
[splitMergesToChildSplit-PivotFail]	*integer*	Split Merge Child Split Pivot.
[splitMergesToParent]	*integer*	Split Merges to Parent.
[splitMergesToParentFail]	*integer*	Split Merge to Parent Fail.
[splitMergesToParentFail-Node]	*integer*	Split Merge to Parent Fail Node.
[splitMergesToParentFail-Tcam]	*integer*	Split Merge to Parent Fail Tcam.
[splitMergesToParentNon-Pivot]	*integer*	Split Merges to Parent Non Pivot.
[splitMergesToParentPivot]	*integer*	Split Merges to Parent Pivot.
splits	*integer*	Splits Stats.
splitsByBucket-Redistribution	*integer*	Splits by Bucket Redistribution.
splitsByCount	*integer*	Splits by Count.
splitsByRedistribution	*integer*	Splits by Redistribution.
splitsFail	*integer*	Splits Fail.
splitsFailTcam	*integer*	Splits Fail Tcam.

2.1473 show platform trident l3 software alpm-table routes-v6 [...]

revision 2

see 2.1472

2.1474 show platform trident l3 software ecmp

simpleUcastEcmpRange	*table(string) = simpleUcastEcmpBlocks* see 2.1474.1	A mapping of ECMP Range to ECMP Blocks.

2.1474.1 simpleUcastEcmpBlocks

ecmpBlockDetail	*list of simpleUcastEcmpBlockDetails* see 2.1474.1.1	A mapping of ECMP Blocks to ECMP group ID.
ecmpFreeEntries **	*integer*	Number of free entries. **Since: 4.23.0
ecmpRangeMaxId	*integer*	Maximum ID of ECMP range.
ecmpRangeMinId	*integer*	Minimum ID of ECMP range.

2.1474.1.1 simpleUcastEcmpBlockDetails

ecmpBlockSize	*integer*	ECMP block size.
[ecmpGrpId]	*integer*	ECMP group ID.
ecmpId	*integer*	ECMP ID.
[shadowEcmpId]	*integer*	ECMP shadow ID.

2.1475 show platform trident l3 software host-table [...]

[bankEntriesList]	*list of HostBankBucketInfo* see 2.1475.1	Status and usage of each host bank entry.
[banks]	*table(numeric string) = HostBankInfo* see 2.1475.2	Host banks, keyed by logical bank ID.
[fwdTableMode]	*integer*	Forwarding Mode.
[hostTableEntries]	*integer*	Entries Allocated.
[hostTableSize]	*integer*	Size of the host table.
[hostsV4]	*integer*	Number of IPv4 unicast hosts.
[hostsV6]	*integer*	Number of IPv6 unicast hosts.
[mroutesV4]	*integer*	Number of IPv4 multicast hosts.
[mroutesV6]	*integer*	Number of IPv6 multicast hosts.

2.1475.1 HostBankBucketInfo

buckets	*table(numeric string) = HostBucketInfo* see 2.1475.1.1	Host buckets, keyed by bucket ID.
[logicalBank]	*integer*	Logical Bank ID.
[physicalBank]	*integer*	Physical Bank ID.
[status]	*string*	Message if bank is unused or no buckets in use.

2.1475.1.1 HostBucketInfo

entries	*table(numeric string) = HostBucketEntryInfo* see 2.1475.1.1.1	Host bucket entries, keyed by entry ID.
state	*string*	Bucket State.

2.1475.1.1.1 HostBucketEntryInfo

bucketType	*enum: v6Uc | v4Uc | v6Mc | v4Mc*	bucket type.
[grp]	*string*	Group.
host	*string*	Host.
hwEntry	*string*	Hardware entry position.
[src]	*string*	Source.
[vlan]	*integer*	VLAN ID.
vrf	*integer*	VRF ID.

2.1475.2 HostBankInfo

hostsV4	*integer*	Number of IPv4 hosts.
hostsV6	*integer*	IPv6 hosts.
mroutesV4	*integer*	IPv4 multicast routes.
physicalBank	*integer*	Physical Bank ID.
v4McastEntries	*integer*	Number of IPv4 multicast routes.

v6UcastEntries	*integer*	Number of IPv6 hosts.

2.1476 show platform trident l3 software interfaces tunnel decap [...]

tunnels	*table(numeric string) = L3ShwL3IntfTunnelEntry* see 2.1476.1	Individual L3 tunnel keyed by tunnel ID.

2.1476.1 L3ShwL3IntfTunnelEntry

destinationIp	*IP address*	Destination IP address.
entryId	*integer*	Table entry index.
grePayloadP4	*integer*	GRE Payload IPv4.
grePayloadP6	*integer*	GRE Payload IPv6.
iif	*integer*	L3 IIF entryId.
ipHeader	*string*	Ip header.
protocol	*string*	Protocol.
sourceIp	*IP address*	Source IP address.
tunnelType	*string*	Tunnel Type.

2.1477 show platform trident l3 software interfaces tunnel encap [...]

tunnels	*table(numeric string) = L3ShwEgrIpIntfTunnelEntry* see 2.1477.1	Individual EGR tunnel keyed by tunnel ID.

2.1477.1 L3ShwEgrIpIntfTunnelEntry

destinationIp	*IP address*	Destination IP address.
dmac **	*MAC address*	Destionation Mac address. **Since: 4.26.2
dscp	*integer*	DSCP.
ecn	*integer*	ECN.
entryId	*integer*	Table entry index.
entryType	*string*	entryType.
sourceIp	*IP address*	Source IP address.
timeToLive	*integer*	Time To Live.
tunnelType	*string*	Tunnel Type.

2.1478 show platform trident l3 software l3-iif-profile

iifProfiles	*table(numeric string) = IifTableRow* see 2.1478.1	A mapping of IIF Profiles, keyed by their index.

2.1478.1 IifTableRow

urpfDefaultRouteCheck	*boolean*	Perform unicast RPF check on default routes in the DEFIP table.
urpfMode	*enum: loose | disable | strict*	Controls the type of unicast RPF checks that are enabled.
v4L3Enable	*boolean*	IPv4 unicast routing is enabled for this SVP/SVP-LAG.
v4McL3Enable	*boolean*	IPv4 multicast routing is enabled from this SVP/SVP-LAG.
v4UnknownMcToCpu	*boolean*	Unknown IPv4 Multicast (IPMC miss) is sent to the CPU.
v6L3Enable	*boolean*	IPv6 unicast routing is enabled for this SVP/SVP-LAG.
v6McL3Enable	*boolean*	IPv6 multicast routing is enabled from this SVP/SVP-LAG.
v6UnknownMcToCpu	*boolean*	Unknown IPv6 Multicast (IPMC miss) is sent to the CPU.

2.1479 show platform trident l3 software mgroups [...]

groups	*table(numeric string) = MGroupModel* see 2.1479.1	A mapping of group ID to multicast groups.

2.1479.1 MGroupModel

[hiGig]	*boolean*	Multicast group has HiGig ports.
modules	*table(numeric string) = PortListModel* see 2.1479.1.1	A mapping of FAP ID to ports in multicast group.
[programmed]	*boolean*	Programmed in hardware.
[removeSGLP]	*boolean*	Remove Source Global Logical Port from Outgoing Interface List.
[tagged]	*boolean*	Group is tagged.

2.1479.1.1 PortListModel

l2Ports	*list of L2PortModel* see 2.1479.1.1.1	L2 ports per multicast group per module ID.
l3Ports	*list of L3PortModel* see 2.1479.1.1.2	L3 ports per multicast group per module ID.

2.1479.1.1.1 L2PortModel

[interface]	*interface name*	Interface.
[logId]	*integer*	Logical port ID.
[mmuId]	*integer*	MMU port ID.
[phyId]	*integer*	Physical port ID.
[port]	*enum: hiGig | allPhyPorts | cpu*	Port type.

2.1479.1.1.2 L3PortModel

interface	*interface name*	Interface.
[logId]	*integer*	Logical port ID.
[mmuId]	*integer*	MMU port ID.
nextHops	*list of NextHopModel* see 2.1479.1.1.2.1	Next hops for L3 port.
[phyId]	*integer*	Physical port ID.
[replListId]	*integer*	Replication list ID.
[switchcardIntf] **	*interface name*	Switchcard Interface. **Since: 4.24.2

2.1479.1.1.2.1 NextHopModel

[nextHopId]	*integer*	Next hop ID.
nextHopTunnel **	*boolean*	Next hop is a tunnel. **Since: 4.24.0

`vlanId`	*integer*	VLAN ID.
`[vlanTagged]`	*boolean*	Next hop is tagged.

2.1480 show platform trident l3 software next-hops [...]

revision 2

[l3SoftwareNextHopsHwHer] **	*L3SoftwareNextHopsHwHer* see 2.1480.1	Head end replication next hop table info. **Since: 4.22.0
[l3SoftwareNextHopsMcast]	*L3SoftwareNextHopsMcast* see 2.1480.2	Multicast next hop table info.
[l3SoftwareNextHopsSdTag]	*L3SoftwareNextHopsSdTag* see 2.1480.3	SdTag next hop table info.
[l3SoftwareNextHopsUcast]	*L3SoftwareNextHopsUcast* see 2.1480.4	Unicast next hop table info.

2.1480.1 L3SoftwareNextHopsHwHer

hwHerNHTable	*table(numeric string) = HwHerNH* see 2.1480.1.1	Dictionary of her next hop identifier to next hop Table.

2.1480.1.1 HwHerNH

[dvpIndex]	*integer*	Hardware destination index.
[interface]	*string*	HER next hop interface name:: Interface name will be full name with exception of the following special names : HiGigLport, VxlanFlood, Recirc, VxlanPeer, EcmpArp, VlanFlood.
macAddress	*MAC address*	HER next hop MAC address.
[moduleId]	*integer*	HER module id.
[moduleIdFab]	*integer*	HER module id for fabric chip.
[moduleIdFp]	*integer*	HER module id for front panel chip.
[nextHopType]	*enum: drop | multicast | localHost | cpuAndDrop | cpuAndDropArpRequired | port | flood | remoteHost | cpuSlow | fp | cpuEcmpArp | cpu | her | flow | lag*	HER next hop type.
[nexthopUnderlay]	*boolean*	Next hop is in underlay.
[nhAdjIndex]	*integer*	HER adjacency index.
[pktRewriteFab]	*boolean*	Status of packet rewrite for fabric chip.
[pktRewriteFp]	*boolean*	Packets are rewritten for front panel chip.
[portId]	*integer*	HER port id.
[portIdFab]	*integer*	HER port info for fabric chip.
[portIdFp]	*integer*	HER port info for front panel chip.
[refCount]	*integer*	HER reference count of the next-hop.
vlanId	*integer*	HER VLAN id.
[vxlanEnabled]	*boolean*	Vxlan is enabled:.

2.1480.2 L3SoftwareNextHopsMcast

mcastVlanIdTable	*table(numeric string) = McastVlanIdTable* see 2.1480.2.1	A mapping of nexthop ID to nexthop entry.

2.1480.2.1 McastVlanIdTable

[interface] **	*interface name*	Multicast next hop interface name. **Since: 4.22.0
refCount	*integer*	Multicast refcount for the next hop.
[taggingEnabled]	*boolean*	Status of tagging for the multicast next hop.
vlanId	*integer*	Multicast Vlan id.

2.1480.3 L3SoftwareNextHopsSdTag

sdTagNextHops	*table(numeric string) = SdTagNextHopTable* see 2.1480.3.1	Dictionary keyed at sdTag next-hop identifier to next-hop table.

2.1480.3.1 SdTagNextHopTable

dvpIndex	*integer*	Hardware destination index.
interface	*interface name*	interface name for the next hop.
[taggingEnabled]	*boolean*	Status of tagging for the SdTag next hop.
vlanId	*integer*	Vlan id.

2.1480.4 L3SoftwareNextHopsUcast

l3SoftwareNextHopsUcastEcmp	*L3SoftwareNextHopsUcastEcmp* see 2.1480.4.1	ECMP Unicast next hop.
l3SoftwareNextHopsUcastSimple	*L3SoftwareNextHopsUcastSimple* see 2.1480.4.2	Simple Unicast next hop Table.

2.1480.4.1 L3SoftwareNextHopsUcastEcmp

ecmpTable	*table(numeric string) = EcmpUcastEcmpId* see 2.1480.4.1.1	Unicast ECMP table dictionary keyed with Ecmp Id.

2.1480.4.1.1 EcmpUcastEcmpId

arpResolverNeeded	*boolean*	Boolean to indicate if Arp resolver is needed.
blockSize	*integer*	Number of ECMP next-hops.
ecmpNextHops	*list of EcmpUcastNextHop* see 2.1480.4.1.1.1	ECMP table list.

ecmpNextHopsDetail	*list of EcmpUcastNextHopDetail* see 2.1480.4.1.1.2	ECMP detail table list.
[nexthopUnderlay]	*boolean*	Flag indicating if the next hop is in underlay.
nhAdjIndex	*integer*	Adjacency index of the ECMP next hop.
refCount	*integer*	Reference count for the ECMP next hop.
startIndex	*integer*	ECMP start index.

2.1480.4.1.1.1 EcmpUcastNextHop

[dvp] **	*integer*	DVP index. **Since: 4.20.0
nextHopId	*integer*	Unicast ECMP next hop id.
[weight]	*integer*	Repetition count/weight of the fec/adjacency.

2.1480.4.1.1.2 EcmpUcastNextHopDetail

bitmapList	*list of integer*	Port bitmap List of ECMP next hop.
[dvp] **	*integer*	DVP index. **Since: 4.20.0
nextHopId	*integer*	Unicast ECMP next hop id.

2.1480.4.2 L3SoftwareNextHopsUcastSimple

simpleUcastNHTable	*table(numeric string) = SimpleUcastNH* see 2.1480.4.2.1	Dictionary of non-ECMP next hop identifier to next hop Table.

2.1480.4.2.1 SimpleUcastNH

[dvpIndex] **	*integer*	Hardware destination index. **Since: 4.18.0
[interface]	*string*	Unicast Non-ECMP next hop interface name:: Interface name will be full name with exception of the following special names : HiGigLport, VxlanFlood, Recirc, VxlanPeer, EcmpArp, VlanFlood.
macAddress	*MAC address*	Unicast Non-ECMP next hop mac address.
[moduleId]	*integer*	Unicast Non-ECMP module id.
[moduleIdFab] **	*integer*	Unicast Non-ECMP module id for fabric chip. **Since: 4.22.0
[moduleIdFp] **	*integer*	Unicast Non-ECMP module id for front panel chip. **Since: 4.22.0

[nextHopType]	*enum: drop* | *multicast* | *localHost* | *cpuAndDrop* | *cpuAndDropArpRequired* | *port* | *flood* | *remoteHost* | *cpuSlow* | *fp* | *cpuEcmpArp* | *cpu* | *her* | *flow* | *lag*	Unicast Non-ECMP next hop type.
[nexthopUnderlay] **	*boolean*	Flag indicating if the next hop is in underlay. **Since: 4.18.0
[nhAdjIndex]	*integer*	Unicast Non-ECMP adjacency index.
[pktRewriteFab] **	*boolean*	Status of packet rewrite for fabric chip. **Since: 4.22.0
[pktRewriteFp] **	*boolean*	Status of packet rewrite for front panel chip. **Since: 4.22.0
[portId]	*integer*	Unicast Non-ECMP port id.
[portIdFab] **	*integer*	Unicast Non-ECMP port info for fabric chip. **Since: 4.22.0
[portIdFp] **	*integer*	Unicast Non-ECMP port info for front panel chip. **Since: 4.22.0
[refCount]	*integer*	Unicast reference count of the next-hop.
vlanId	*integer*	Unicast Non-ECMP Vlan id.
[vxlanEnabled] **	*boolean*	Vxlan is enabled:. **Since: 4.22.0

2.1481 show platform trident l3 software routes [...]

[cmdErr] **	*string*	Error Message. **Since: 4.20.0
[hostTable]	*StrataL3Vrfs* see 2.1481.1	Host route table info.
[lpmTable]	*StrataL3Vrfs* see 2.1481.1	LPM route table info.

2.1481.1 StrataL3Vrfs

vrfs	*table(numeric string) = RoutesPerVrfEntry* see 2.1481.1.1	A mapping of vrf id to routes.

2.1481.1.1 RoutesPerVrfEntry

routes	*table(string) = StrataL3RouteEntry* see 2.1481.1.1.1	A mapping of prefix to route entry.

2.1481.1.1.1 StrataL3RouteEntry

classId	*integer*	Class ID marking for packets in ingress pipeline.
ecmp	*boolean*	Flag indicating if route has ECMP nexthops.
[ecmpGroupId]	*integer*	ECMP nexthop ID.
[ecmpGroupSize]	*integer*	ECMP group size.
entryId	*integer*	Entry ID.
[hit]	*boolean*	Flag indicating if the traffic hit the route.
[nexthopId]	*integer*	Nexthop ID.
[nexthops]	*table(numeric string) = SimpleUcastNH* see 2.1480.4.2.1	A mapping of nexthop ID to nexthop.
[position]	*integer*	Position in the TCAM table.
[subBucket] **	*integer*	Indicate sub bucket pointer in shared SRAM. **Since: 4.22.1

2.1482 show platform trident l3 software routes-v6 [...]

see 2.1481

2.1483 show platform trident l3 software routing-vlan [...]

perVlanEntries	*list of PerVlanEntry* see 2.1483.1	List of per vlan routing entries.

2.1483.1 PerVlanEntry

entryId	*string*	Entry id.
macAddress	*MAC address*	Mac address.
macAddressMask	*MAC address*	Mac adress mask.
vlanId	*integer*	Vlan id.
vlanMask	*string*	Vlan mask.

2.1484 show platform trident l3 software urpf exception-list [...]

unsupportedRules	*list of UnsupportedAclRule* see 2.1484.1	ACL rules not supported by hardware.
urpfExceptions	*list of UrpfException* see 2.1484.2	uRPF exceptions.

2.1484.1 UnsupportedAclRule

aclName	*string*	ACL name.
aclType	*enum: IPv6 | IPv4*	ACL type.
unsupportedRuleId	*integer*	ACL rule id, not supported by hardware.

2.1484.2 UrpfException

aclName	*string*	ACL name.
aclType	*enum: IPv6 | IPv4*	ACL type.
[fapId] **	*integer*	Chip FAP ID. **Since: 4.24.2
hardwareStatus	*enum: Not Programmed | Programmed*	Programmed in hardware.
interface	*interface name*	Interface.
[memberInterface]	*interface name*	Member interface.
[modId]	*integer*	Chip module id (legacy attribute, use fapId instead).

2.1485 show platform trident l3 summary

revision 3

[alpm]	*AlpmTableState* see 2.1472.1	ALPM table.
configRsvdHostsV4	*integer*	Number of configured and reserved IPv4 hosts.
configRsvdHostsV6	*integer*	Number of configured and reserved IPv6 hosts.
configRsvdMcastRoutesV4	*integer*	Number of configured and reserved IPv4 multicast routes.
configRsvdMcastRoutesV6	*integer*	Number of configured and reserved IPv6 multicast routes.
fapType	*string*	Front panel chip model.
fapTypesFabric	*list of string*	List of fabric chip models.
fastdropMcastRoutes	*integer*	Number of fastdrop multicast routes.
forwardingMode	*string*	Forwrading mode.
hostMcastRoutesV4	*integer*	Number of IPv4 multicast routes.
hostMcastRoutesV6	*integer*	Number of IPv6 multicast routes.
hostTableMode	*integer*	Host table mode.
hostsV4	*integer*	Number of IPv4 hosts.
hostsV6	*integer*	Number of IPv6 hosts.
ingDvpEntries	*integer*	Number of ingress DVP table entries in use.
ipTunnels	*integer*	Number of IP tunnels in use.
ipV4Routes	*integer*	Number of IPv4 routes (hosts+lpm).
ipV6Routes	*integer*	Number of IPv6 routes (hosts+lpm).
l2mcTablesizeL2	*integer*	L2MC table size L2.
l2mcTablesizeL3	*integer*	L2MC table size L3.
l3Intf	*integer*	Number of L3 interfaces.
lpmEntriesV4	*integer*	Number of LPM IPv4 entries in use.
lpmEntriesV6Full	*integer*	Number of LPM IPv6 full entries in use.
lpmEntriesV6Half	*integer*	Number of LPM IPv6 half entries in use.
lpmHalfFullEntries	*integer*	Number of LPM half full entries in use.
lpmMode	*integer*	LPM table mode.
maxEcmpRoutes	*integer*	Maximum possible number of ECMP paths/routes.
maxHosts	*integer*	Maximum possible number of IPv4 and IPv6 hosts.
maxIngDvpEntries	*integer*	Maximum possible number of ingress DVP table entries.
maxIpTunnels	*integer*	Maximum possible number of IP tunnels.
maxL3Intf	*integer*	Maximum possible number of L3 interfaces.
maxLpmEntries	*integer*	Maximum possible number of LPM IPv4 and IPv6 entries.
maxMcastGroups	*integer*	Maximum possible number of multicast replication groups.
maxMyStationTcamTable-Entries	*integer*	Maximum number of MyStationTcamTable entries.

[maxNextHopEcmpEntries]	*integer*	Maximum possible number of NextHop ECMP entries in Trident2.
[maxNextHopEcmpSets]	*integer*	Maximum possible number of NextHop ECMP groups in Trident2.
maxNextHopEntries	*integer*	Maximum possible number of NextHop unicast/multicast entries.
mcastGroups	*integer*	Number of multicast replication groups in use.
mcastModules	*list of McastModule* see 2.1485.1	Multicast modules.
mcastRoutesV4	*integer*	Number of IPv4 multicast routes.
mcastRoutesV6	*integer*	Number of IPv6 multicast routes.
mcastUnprogrammedRoutes	*table(string) = integer*	Multicast unprogrammed IPv4 routesstatistics. Key is VRF, value is the numberof unprogrammed multicast routes in thatVRF.
mcastUnprogrammedRoutesV6	*table(string) = integer*	Multicast unprogrammed IPv6 routesstatistics. Key is VRF, value is the numberof unprogrammed multicast routes in thatVRF.
mcastV4Enabled	*boolean*	Status of IPv4 multicast routing.
mcastV6Enabled	*boolean*	Status of IPv6 multicast routing.
mplsEnabled	*boolean*	Status of MPLS routing.
myStationTcamTableEntries	*integer*	Number of MyStationTcamTable entries in use.
[nextHopEcmpEntries]	*integer*	Number of NextHop ECMP entries in Trident2.
[nextHopEcmpSets]	*integer*	Number of NextHop ECMP groups in use in Trident2.
nextHopL3MCEntries	*integer*	Number of NextHop L3MC entries in use.
nextHopMcastEntries	*integer*	Number of NextHop multicast entries in use.
nextHopUcastEntries	*integer*	Number of NextHop unicast entries in use.
numUnprogrammedVrrp	*integer*	Unprogrammed VRRP group count.
[overlayInterfaceEntries]	*integer*	Number of interface entries in Overlay region.
[overlayInterfaceEntries-Reserved]	*integer*	Number of interface entries reserved for the Overlay region.
[overlayMaxNextHopEcmp-Entries]	*integer*	Maximum possible number of NextHop ECMP entries in Overlay region.
[overlayMaxNextHopEcmpSets]	*integer*	Maximum possible number of NextHop ECMP groups in Overlay region.
[overlayNextHopEcmpEntries]	*integer*	Number of NextHop ECMP entries in Overlay region.
[overlayNextHopEcmpEntries-MaxId]	*integer*	Maximum entry Id of ECMP Nexthops in Overlay region.
[overlayNextHopEcmpEntries-MinId]	*integer*	Minimum entry Id of ECMP Nexthops in Overlay region.
[overlayNextHopEcmpSets]	*integer*	Number of NextHop ECMP groups in usein Overlay region.
[overlayNextHopEcmpSetsMax-Id]	*integer*	Maximum group Id of NextHop ECMP groups in Overlay region.

[overlayNextHopEcmpSetsMinId]	*integer*	Minimum group Id of NextHop ECMP groups in Overlay region.
[overlayNexthopBanks]	*integer*	Number of banks allocated to overlay Nexthops.
[overlayNexthopEntries]	*integer*	Number of NextHop entries in overlay region.
[overlayNexthopEntriesReserved]	*integer*	Number of NextHop entries reserved for the overlay region.
[perVlanRoutingEnabled]	*boolean*	Status of per vlan routing.
platformAdjSharing	*boolean*	Status of Platform Adjacency sharing.
reservedMcastGroups	*integer*	Number of reserved multicast replication groups in use.
routeDeletionDelay	*floating point number*	Route deletion delay.
rsvdHostsV4	*integer*	Number of reserved IPv4 hosts.
rsvdHostsV6	*integer*	Number of reserved IPv6 hosts.
rsvdMcastRoutesV4	*integer*	Number of reserved IPv4 multicast routes.
rsvdMcastRoutesV6	*integer*	Number of reserved IPv6 multicast routes.
[sdtagNexthopEntries]	*integer*	Number of SDTag Nexthop entries.
ucastV4Enabled	*boolean*	Status of IPv4 unicast routing.
ucastV6Enabled	*boolean*	Status of IPv6 unicast routing.
[underlayInterfaceEntries]	*integer*	Number of interface entries in Underlay region.
[underlayInterfaceEntriesReserved]	*integer*	Number of interface entries reserved for the Underlay region.
[underlayMaxNextHopEcmpEntries]	*integer*	Maximum possible number of NextHop ECMP entries in Underlay region.
[underlayMaxNextHopEcmpSets]	*integer*	Maximum possible number of NextHop ECMP groups in Underlay region.
[underlayNextHopEcmpEntries]	*integer*	Number of NextHop ECMP entries in Underlay region.
[underlayNextHopEcmpEntriesMaxId]	*integer*	Maximum entry Id of ECMP Nexthops in Underlay region.
[underlayNextHopEcmpEntriesMinId]	*integer*	Minimum entry Id of ECMP Nexthops in Underlay region.
[underlayNextHopEcmpSets]	*integer*	Number of NextHop ECMP groups in use in Underlay region.
[underlayNextHopEcmpSetsMaxId]	*integer*	Maximum group Id of NextHop ECMP groups in Underlay region.
[underlayNextHopEcmpSetsMinId]	*integer*	Minimum group Id of NextHop ECMP groups in Underlay region.
[underlayNexthopBanks]	*integer*	Number of banks allocated to underlay Nexthops.
[underlayNexthopEntries]	*integer*	Number of NextHop entries in underlay region.
[underlayNexthopEntriesReserved]	*integer*	Number of NextHop entries reserved for the underlay region.
unprogrammedV4Routes	*integer*	Number of unprogrammed IPv4 routes.
unprogrammedV6Routes	*integer*	Number of unprogrammed IPv6 routes.
urpfEnabled	*boolean*	Status of uRPF.

2.1485.1 McastModule

module	*integer*	Module ID.
[replHeadEntries]	*integer*	Number of replication head entries in use.
replHeadEntriesPerPipe	*table(string) = integer*	ReplHead entries in use per pipe.
replHeadMaxEntries	*integer*	Maximum number of replication head entries.
[replListEntries]	*integer*	Number of replication list entries in use.
replListEntriesPerPipe	*table(string) = integer*	ReplList entries in use per pipe.
replListMaxEntries	*integer*	Maximum number of replication list entries.

2.1486 show platform trident lag [...]

lags	*table(string) = LagDetailSingleEntry* see 2.1486.1	LAG details keyed by LAG ID.

2.1486.1 LagDetailSingleEntry

inHardware	*boolean*	Hardware or software data is stored in.
[interfaces]	*list of LagDetailSingleInterfaceEntry* see 2.1486.1.1	Interfaces in the LAG.
ports	*integer*	Number of ports in LAG stored in software.
[portsHardware]	*integer*	Number of ports in LAG stored in hardware.
[trunkHardware]	*integer*	LAG ID in hardware.

2.1486.1.1 LagDetailSingleInterfaceEntry

[interface]	*string*	Front panel interface ID.
[modId]	*integer*	Switchcard module ID in software.
[modIdHardware]	*integer*	Switchcard module ID in hardware.
[portId]	*integer*	Switchcard port ID in software.
[portIdHardware]	*integer*	Switchcard port ID in hardware.

2.1487 show platform trident lag detail [...]

lags	*table(string) = LagDetailSingleEntry* see 2.1486.1	LAG details keyed by LAG ID.

2.1488 show platform trident lag nonucast-trunk-block-mask l3mc [...]

linecards	*table(string) = NonUcastTrunkBlockMaskTbl* see 2.1488.1	NonUcastTrunkBlockMask table grouped by linecards.

2.1488.1 NonUcastTrunkBlockMaskTbl

entries	*table(numeric string) = NonUcastTrunkBlockMaskEntry* see 2.1488.1.1	NonUcastTrunkBlockMask table entries.

2.1488.1.1 NonUcastTrunkBlockMaskEntry

interfaces	*list of string*	Interfaces in NonUcastTrunkBlockMask entry.

2.1489 show platform trident lag software

currentLagType	*enum: hardware | software*	Current LAG Type.
initialLagType	*enum: hardware | software*	Initial LAG Type.
transitionType	*enum: delayed | immediate*	Transition Type.

2.1490 show platform trident lanz coprocessor status

chips	*table(string) = CopStatusEntry* see 2.1490.1	Coprocessor status keyed by chip name.

2.1490.1 CopStatusEntry

chipInfoEntryList	*list of ChipInfoEntry* see 2.1490.1.1	List of Chip info entry.
configEntryList	*list of ConfigEntry* see 2.1490.1.2	List of Config entry.
csrEntryList	*list of CsrEntry* see 2.1490.1.3	List of CSR entry.
statusEntryList	*list of StatusEntry* see 2.1490.1.4	List of Status entry.
timeSyncEntryList	*list of TimeSyncEntry* see 2.1490.1.5	List of Time sync entry.

2.1490.1.1 ChipInfoEntry

cpuPort	*integer*	Cpu Port id.
portQueueInfo	*table(numeric string) = PortQueueInfoEntry* see 2.1490.1.1.1	Dict of Port Queue Info keyed by MMU port number.

2.1490.1.1.1 PortQueueInfoEntry

mCastQBase	*integer*	Start of Multicast queue.
mCastQCount	*integer*	Count of Multicast queue.
uCastQBase	*integer*	Start of Unicast queue.
uCastQCount	*integer*	Count of Unicast queue.

2.1490.1.2 ConfigEntry

cpuPortProfiles	*table(numeric string) = HighLowCellEntry* see 2.1490.1.2.1	CPU port profile dict keyed by port number.
dmaRecords	*integer*	Number of DMA records.
dmaSize	*integer*	Size of DMA buffer in bytes.
ethernetFabricPortProfiles	*table(numeric string) = HighLowCellEntry* see 2.1490.1.2.1	Port profile dict keyed by port number.
mirroringDisabled	*table(numeric string) = integer*	MMU port Entry dict keyed by port number.
mtpAddress	*string*	EMIRROR_CONTROLX address for mirroring.
numUpdAgg	*integer*	Number of aggressive update messages.
portConfig	*table(numeric string) = integer*	Port config dict keyed by port number.
qlenUpdAggInterval	*integer*	Time between aggressiveupdate messages in microseconds.

qlenUpdInterval	*integer*	Time between update messages in microseconds.

2.1490.1.2.1 HighLowCellEntry

highCells	*integer*	High cells.
lowCells	*integer*	Low cells.

2.1490.1.3 CsrEntry

crCfgChg	*boolean*	Config register: config needs to be updated.
crEnable	*boolean*	Config register: toggles LANZ.
crStatusUpdate	*integer*	Config register: Status was updated.
crTrace	*boolean*	Config register: toggles tracing.
dmaHeadPtr	*string*	DMA head pointer.
dmaReadPtr	*string*	Position of the DMA read pointer.
srCfgChg	*boolean*	Status Register: config was updated.
srInitialized	*boolean*	Status register: indicates run status.
srStatusUpdate	*integer*	Status Register: Status was updated.

2.1490.1.4 StatusEntry

dmaWritePtr	*string*	Position of last DMA msg written in bytes.
failedTransferCnt	*integer*	Number of failed transfers.
magicNumber	*string*	Should be 0x4c414e5a.
timestampH	*string*	High order bits of timestamp.
timestampL	*string*	Low order bits of timestamp.

2.1490.1.5 TimeSyncEntry

[hwTs]	*integer*	Current hardware timestamp.
[hwTsToUSec]	*string*	Current hardware timestamp in microseconds.
[noHwTsError]	*string*	Error String when unable to get Timestamp.
[now]	*string*	Current time in UTC.
[numTsUpdateErrors]	*integer*	Total number of errors encountered.
[numTsUpdates]	*integer*	Total number of timestamp updates.
[offset]	*integer*	Time offset in nanoseconds.
[state]	*enum: 1 | 0 | 2 | 3*	Current state of TimeSync: 0 : unknown1 : synchronizing2 : backOff3 : errorDetected.

2.1491 show platform trident lanz hardware status

chips	*table(string) = HwStatusEntry* see 2.1491.1	Hardware Status keyed by chip name.

2.1491.1 HwStatusEntry

freeBstProfiles	*integer*	Number of available new thresholds for Ethernet ports.
freeBstProfilesCpu	*integer*	Number of available new thresholds for Cpu port.
freeBstProfilesFabric	*integer*	Number of available new thresholds for Fabric ports.
interfaces	*table(Interface name) = SingleInterfaceEntry* see 2.1491.1.1	Interface profiles keyed by interface names.
thresholdToBstProfiles	*integer*	Number of thresholds in use for Ethernet ports.
thresholdToBstProfilesCpu	*integer*	Number of thresholds in use for Cpu port.
thresholdToBstProfiles-Fabric	*integer*	Number of thresholds in use for Fabric ports.

2.1491.1.1 SingleInterfaceEntry

highThreshold	*string*	High Threshold.
hwProfile	*string*	Hardware Profile, * Unavailable.

2.1492 show platform trident mac-address-table [...]

see 2.1425

2.1493 show platform trident mac-address-table diff [...]

see 2.1426

2.1494 show platform trident mac-address-table missing [...]

see 2.1427

2.1495 show platform trident mac-traffic-policy hardware-id summary

matchRules	*table(string) = MatchRule-HardwareSummary* see 2.1495.1	A mapping of rule name to hardware ID information.

2.1495.1 MatchRuleHardwareSummary

[aclName]	*string*	Access list name.
[hardwareId]	*integer*	Hardware ID for access list.

2.1496 show platform trident mac-traffic-policy rules status [...]

chips	*table(string) = MacTrafficPolicyRulesChipStatus* see 2.1496.1	A mapping of chip name to status of MAC traffic policy rules.

2.1496.1 MacTrafficPolicyRulesChipStatus

rules	*table(string) = MacTrafficPolicyRuleStatus* see 2.1496.1.1	A mapping of rule name to operational status information.

2.1496.1.1 MacTrafficPolicyRuleStatus

status	*enum: success | inProgress | notApplicable | failed*	Status type.

2.1497 show platform trident mirror index [...]

modules	*table(numeric string) = IndexTable* see 2.1497.1	Dictionary mapping a module ID to the index entries.

2.1497.1 IndexTable

free	*integer*	Free MTP indices.
indexes	*list of IndexRow* see 2.1497.1.1	Index usage information.
totalMtpIndices	*integer*	Total MTP indices.

2.1497.1.1 IndexRow

direction	*string*	Mirror direction.
index	*integer*	Index.
mirrorInterface	*interface name*	Mirror interface.
refs	*integer*	Reference count.

2.1498 show platform trident mirror session [...]

sessions	*table(string) = SessionInfo* see 2.1498.1	Dictionary mapping session name to details.

2.1498.1 SessionInfo

index	*list of SessionInfoRow* see 2.1498.1.1	List of port indices and module ID.
targetIntf	*interface name*	Target interface.

2.1498.1.1 SessionInfoRow

egressIndex	*integer*	Egress port index.
ingressIndex	*integer*	Ingress port index.
module	*integer*	Module ID.

2.1499 show platform trident mmu buffers status

moduleInfo	*table(string) = StrataMmuBufferModel* see 2.1499.1	A mapping between a linecard and MMU data.

2.1499.1 StrataMmuBufferModel

mmuData	*table(numeric string) = StrataMmuInfo* see 2.1499.1.1	A mapping between a MMU ID and MMU data.

2.1499.1.1 StrataMmuInfo

egressSharedCount	*integer*	Egress shared buffer cell usage count.
egressSharedTotal	*integer*	Egress shared buffer cell total count.
egressUsage	*integer*	Egress shared buffer cell usage percentage.
ingressHeadroomCount	*integer*	Ingress Headroom buffer cell usage count.
ingressHeadroomTotal	*integer*	Ingress Headroom buffer cell total count.
ingressHeadroomUsage	*integer*	Ingress Headroom buffer cell usage percentage.
ingressSharedCount	*integer*	Ingress shared buffer cell usage count.
ingressSharedTotal	*integer*	Ingress shared buffer cell total count.
ingressUsage	*integer*	Ingress shared buffer cell usage percentage.

2.1500 show platform trident mmu queue cpu-profile [...]

[availableProfiles]	*list of string*	Available CPU queue profiles.
profileName	*string*	Profile name.
queues	*table(string) = CpuQueueConfig* see 2.1500.1	Queue configs, keyed by their name.

2.1500.1 CpuQueueConfig

reserved	*Reserved* see 2.1500.1.1	The reserved value for this queue.
threshold	*Threshold* see 2.1500.1.2	The threshold value for this queue.

2.1500.1.1 Reserved

isDefault	*boolean*	The reserved value is the system default.
unit	*enum: bytes | cells*	Unit for the reserved value.
value	*integer*	The amount of memory to be reserved.

2.1500.1.2 Threshold

isDefault	*boolean*	The threshold value is the system default.
value	*floating point number*	Dynamic shared memory threshold.

2.1501 show platform trident mmu queue headroom

interfaces	*table(Interface name) = integer*	Available MMU headroom (Cells) for each priority group of an interface.
[poolLimit]	*integer*	Ingress headroom pool size (Cells) for each MMU.

2.1502 show platform trident mmu queue status [...]

revision 2

interfaces	*table(Interface name) = PlatformInterfaceMmuStatusModel* see 2.1502.1	Mapping between interfaces and associated MMUs.

2.1502.1 PlatformInterfaceMmuStatusModel

egressMmuStatus	*table(numeric string) = PlatformInterfaceEgressMmuStatusModel* see 2.1502.1.1	Mapping between egress MMU IDs and egress queue status.
ingressMmuStatus	*table(numeric string) = PlatformInterfaceIngressMmuStatusModel* see 2.1502.1.2	Mapping between ingress MMU IDs and priority group status.
[numMmu] **	*integer*	Total number of MMUs in the chip. **Since: 4.21.3

2.1502.1.1 PlatformInterfaceEgressMmuStatusModel

mcastQueues	*table(numeric string) = PlatformMmuStatusModel* see 2.1502.1.1.1	Mapping between multicast queue IDs and MMU status.
ucastQueues	*table(numeric string) = PlatformMmuStatusModel* see 2.1502.1.1.1	Mapping between unicast queue IDs and MMU status.

2.1502.1.1.1 PlatformMmuStatusModel

[reserved]	*integer*	Queue minimum reserved bytes.
[threshold]	*floating point number*	Queue threshold bytes.
used	*integer*	Queue used bytes.

2.1502.1.2 PlatformInterfaceIngressMmuStatusModel

priorityGroups	*table(numeric string) = PlatformMmuStatusModel* see 2.1502.1.1.1	Mapping between priority group IDs and MMU status.

2.1503 show platform trident mmu queue status cpu [...]

slices	*table(string) = PlatformCpuStatusModel* see 2.1503.1	Mapping from slice to associated CPU port status model.

2.1503.1 PlatformCpuStatusModel

egressMmus	*table(numeric string) = PlatformCpuEgressMmuStatusModel* see 2.1503.1.1	Mapping between egress MMU IDs and egress queue status.
ingressMmus	*table(numeric string) = PlatformInterfaceIngressMmuStatusModel* see 2.1502.1.2	Mapping between ingress MMU IDs and priority group status.
[numMmu]	*integer*	Total number of MMUs in the chip.

2.1503.1.1 PlatformCpuEgressMmuStatusModel

cpuQueues	*table(numeric string) = PlatformMmuStatusModel* see 2.1502.1.1.1	Mapping between CPU queue IDs and MMU status.

2.1504 show platform trident mmu queue watermark [...]

`[mmuIngressHeadroomPool-Watermark]` **	*StrataMmuIngress-HeadroomPoolWatermark-Model* see 2.1504.4	MMU Ingress Headroom Pool Watermarks (Cells) for all MMUs. **Since: 4.18.0
`mmuIngressHeadroomWatermark`	*StrataInterfaceMmuIngress-WatermarkModel* see 2.1504.1	MMU Ingress Headroom Watermarks (Cells) for all interfaces.
`mmuIngressSharedWatermark`	*StrataInterfaceMmuIngress-WatermarkModel* see 2.1504.1	MMU Ingress Shared Watermarks (Cells) for all interfaces.
`mmuMulticastSharedWatermark`	*StrataInterfaceMmu-MulticastSharedWatermark-Model* see 2.1504.2	MMU Multicast Shared Watermarks (Cells) for all interfaces.
`mmuUnicastSharedWatermark`	*StrataInterfaceMmu-UnicastSharedWatermark-Model* see 2.1504.3	MMU Unicast Shared Watermarks (Cells) for all interfaces.
`[numMmu]` **	*integer*	Total number of MMUs in the chip. **Since: 4.21.3

2.1504.1 StrataInterfaceMmuIngressWatermarkModel

`interfaces`	*table(Interface name) = StrataInterfaceMmuIngress-WatermarkMMUModel* see 2.1504.1.1	Mapping from interface to associated ingress MMUs for MMU Ingress Headroom/Shared Watermarks.

2.1504.1.1 StrataInterfaceMmuIngressWatermarkMMUModel

`mmus`	*table(numeric string) = StrataInterfaceMmuQueue-WatermarkPGModel* see 2.1504.1.1.1	Mapping from ingress MMU IDs to priority groups watermarks.

2.1504.1.1.1 StrataInterfaceMmuQueueWatermarkPGModel

`priorityGroups`	*table(numeric string) = integer*	Mapping from priority group number (0-7) to MMU Ingress (Headroom/Shared) Watermarks (Cells).

2.1504.2 StrataInterfaceMmuMulticastSharedWatermarkModel

`interfaces`	*table(Interface name) = StrataInterfaceMmu-MulticastSharedWatermark-MMUModel* see 2.1504.2.1	Mapping from interface to associated egress MMUs for MMU Multicast Shared Watermarks.

2.1504.2.1 StrataInterfaceMmuMulticastSharedWatermarkMMUModel

mmus	*table(numeric string) = StrataInterfaceMmuQueueWatermarkMCModel* see 2.1504.2.1.1	Mapping from egress MMU IDs to multicast queue watermarks.

2.1504.2.1.1 StrataInterfaceMmuQueueWatermarkMCModel

queues	*table(numeric string) = integer*	Mapping from multicast queue number (0-7) to MMU Multicast Shared Watermarks (Cells).

2.1504.3 StrataInterfaceMmuUnicastSharedWatermarkModel

interfaces	*table(Interface name) = StrataInterfaceMmuUnicastSharedWatermarkMMUModel* see 2.1504.3.1	Mapping from interface to associated egress MMUs for MMU Unicast Shared Watermarks.

2.1504.3.1 StrataInterfaceMmuUnicastSharedWatermarkMMUModel

mmus	*table(numeric string) = StrataInterfaceMmuQueueWatermarkUCModel* see 2.1504.3.1.1	Mapping from egress MMU IDs to unicast queue watermarks.

2.1504.3.1.1 StrataInterfaceMmuQueueWatermarkUCModel

queues	*table(numeric string) = integer*	Mapping from unicast queue number (0-7) to MMU Unicast Shared Watermarks (Cells).

2.1504.4 StrataMmuIngressHeadroomPoolWatermarkModel

units	*table(numeric string) = StrataUnitMmuIngressHeadroomPoolWatermarkModel* see 2.1504.4.1	Mapping from Switch units (chips) to associated ingress MMUs for MMU Ingress Headroom/Shared Watermarks.

2.1504.4.1 StrataUnitMmuIngressHeadroomPoolWatermarkModel

mmus	*table(numeric string) = integer*	Mapping from ingress MMU IDs to Headroom Pool Watermarks (Cells).

2.1505 show platform trident mmu queue watermark cpu [...]

slices	*table(string) = StrataCpuMmuQueueWatermarkModel* see 2.1505.1	Mapping from linecard to associated CPU port MMU Queue Watermarks.

2.1505.1 StrataCpuMmuQueueWatermarkModel

[cpuIngressSharedWatermark]	*StrataInterfaceMmuIngressWatermarkMMUModel* see 2.1504.1.1	MMU Ingress Shared Watermarks (Cells) for CPU port.
[cpuMulticastSharedWatermark]	*StrataInterfaceMmuMulticastSharedWatermarkMMUModel* see 2.1504.2.1	MMU Multicast Shared Watermarks (Cells) for CPU port.

2.1506 show platform trident parity-event [...]

chipParityInformation	*table(string) = ChipParityInformation* see 2.1506.1	Parity information for a chip, keyed by chip name.
parityErrorCount	*integer*	Total parity error count.

2.1506.1 ChipParityInformation

fapId **	*integer*	Unique numerical chip identifier. **Since: 4.25.0
memParityErrors	*table(string) = MemParityInformation* see 2.1506.1.1	Parity information for a specific memory on the chip, keyed by memory name.
modId	*integer*	Chip module ID (legacy attribute, use fapId instead.
parityErrorCount	*integer*	Total parity error count on the chip.
sliceName	*string*	Name of the slice managing the chip.

2.1506.1.1 MemParityInformation

memId	*integer*	Unique numerical table identifier.
memType	*enum: other | reg | table*	The type of the memory, can be either a table or register.
offsetParityErrors	*table(numeric string) = OffsetParityInformation* see 2.1506.1.1.1	Parity information for a specific table offset, keyed by offset (int).
parityErrorCount	*integer*	Total number of parity errors reported on this memory.

2.1506.1.1.1 OffsetParityInformation

lastSeen	*floating point number*	Most recent parity event on this offset as seconds since epoc.
parityErrorCount	*integer*	Total number of parity errors reported at this offset.

2.1507 show platform trident priority-flow-control watchdog snapshot [...]

intfSnapshot	*table(Interface name) = IntfSnapshotModel* see 2.1507.1	A mapping of interface name to Snapshot.
snapshotSummary	*SnapshotSummaryModel* see 2.1507.2	Stage Snapshot Summary.

2.1507.1 IntfSnapshotModel

recoveryStageSnapshot	*StageSnapshotModel* see 2.1507.1.1	Recovery Stage Snapshot.
stuckStageSnapshot	*StageSnapshotModel* see 2.1507.1.1	Stuck Stage Snapshot.

2.1507.1.1 StageSnapshotModel

disableQ	*SnapshotModel* see 2.1507.1.1.1	Snapshot for Stage during disable queueing state.
pfcProfile	*SnapshotModel* see 2.1507.1.1.1	Snapshot for Stage during pfc profile state.
updateTcam	*SnapshotModel* see 2.1507.1.1.1	Snapshot for Stage during updating Tcam state.

2.1507.1.1.1 SnapshotModel

requestTime	*floating point number*	Time at which request was sent.
responseTime	*floating point number*	Time at which response was sent.

2.1507.2 SnapshotSummaryModel

recoveryStageSummary	*StageSummaryModel* see 2.1507.2.1	Recovery Stage Snapshot Summary.
stuckStageSummary	*StageSummaryModel* see 2.1507.2.1	Stuck Stage Snapshot Summary.

2.1507.2.1 StageSummaryModel

disableQ	*SummaryModel* see 2.1507.2.1.1	Summary for Stage during disable queueing state.
pfcProfile	*SummaryModel* see 2.1507.2.1.1	Summary for Stage during pfc profile state.
updateTcam	*SummaryModel* see 2.1507.2.1.1	Summary for Stage during updating Tcam state.

2.1507.2.1.1 SummaryModel

count	*integer*	Number of times the interfaces were in this particular stage and state.

maxDuration	*floating point number*	Maximum duration for which the interfaces were in this particular stage and state in milliseconds.
minDuration	*floating point number*	Minimum duration for which the interfaces were in this particular stage and state in milliseconds.
movingAverage	*floating point number*	Moving average of duration for which the interfaces were in this particular stage and state in milliseconds.

2.1508 show platform trident profile [...]

[activeScaleMode]	*TileScale* see 2.1508.2	Active scale for current chip and profile.
chipProfiles	*table(string) = ChipProfileModel* see 2.1508.1	Dictionary of profile information per chip.
systemProfile	*enum: default | Adna*	Current system profile.

2.1508.1 ChipProfileModel

activeProfile	*string*	Active profile configured on the chip.
chipName	*string*	Chip name.
supportedProfiles	*list of string*	Profiles supported by the chip.

2.1508.2 TileScale

[forwarding]	*ForwardingScale* see 2.1508.2.1	Forwarding Scale.

2.1508.2.1 ForwardingScale

[l2]	*FeatureScale* see 2.1508.2.1.1	Layer2 MAC Scale.
[l3Multicast]	*FeatureScale* see 2.1508.2.1.1	Layer3 IP Multicast Scale.
[l3Unicast]	*FeatureScale* see 2.1508.2.1.1	Layer3 IP Unicast Scale.
[num]	*integer*	Mode Number.
vxlanBridging	*boolean*	VxLAN Bridging enabled.
vxlanRouting	*boolean*	VxLAN Routing enabled.

2.1508.2.1.1 FeatureScale

maxScale	*integer*	Maximum Value.
minScale	*integer*	Minimum Value.

2.1509 show platform trident ptp hardware-sync [...]

see 2.1242

2.1510 show platform trident security

common	*CommonStatusModel* see 2.1510.1	Common MACsec information.
egress	*EgressStatusModel* see 2.1510.2	Egress MACsec information.
ingress	*IngressStatusModel* see 2.1510.3	Ingress MACsec information.

2.1510.1 CommonStatusModel

mode	*integer*	Security Association Mode.

2.1510.2 EgressStatusModel

errorStatus	*string*	Egress error status.

2.1510.3 IngressStatusModel

errorStatus	*string*	Ingress error status.
scTcamMiss	*integer*	Ingress SC TCAM miss.
spTcamMiss	*integer*	Ingress SP TCAM miss.

2.1511 show platform trident security interface INTF

Could also be an instance of:
ShowSecIntfStatusDetailModel 2.1511.6

interface	*interface name*	Interface name.
l2Feature	*L2Model* see 2.1511.1	L2 status information.
l3Feature	*L3Model* see 2.1511.2	L3 status information.
pktDrvFeature	*PktDrvModel* see 2.1511.3	PktDrv status information.
securityEngineFeature	*SecurityEngineModel* see 2.1511.4	Security Engine status information.
sliceFeature	*SliceModel* see 2.1511.5	Slice status information.
tcamFeature	*TcamModel* see 2.1511.6	TCAM status information.

2.1511.1 L2Model

[flexL2EntryIndex]	*integer*	Flex L2 Entry Index.
status	*string*	Configuration status.

2.1511.2 L3Model

[mmuReplicationGroups]	*list of string*	MMU Replication Groups.
status	*string*	Configuration status.

2.1511.3 PktDrvModel

status	*string*	Configuration status.

2.1511.4 SecurityEngineModel

[esecSaIndex]	*integer*	Encryption active SA index.
[esecScIndex]	*integer*	Encryption SC index.
[isecSaIndexes]	*string*	Decryption active SA indexes.
[isecScIndex]	*integer*	Decryption SC index.
status	*string*	Configuration status.

2.1511.5 SliceModel

status	*string*	Configuration status.
[subportId]	*integer*	Subport ID.

2.1511.6 TcamModel

[decryptFirstPassEfpEntry]	*integer*	Decryption First pass EFP Entry.
[decryptFirstPassIfpEntry]	*integer*	Decryption First pass IFP Entry.

[decryptSecondPassIfpEntry]	*integer*	Decryption Second pass IFP Entry.
[encryptFirstPassEfpEntry]	*integer*	Encryption First pass EFP Entry.
[encryptSecondPassIfpEntry]	*integer*	Encryption Second pass IFP Entry.
status	*string*	Configuration status.
interface	*interface name*	Interface name.
l2Feature	*L2Model* see 2.1511.1	L2 status information.
l3Feature	*L3Model* see 2.1511.2	L3 status information.
pktDrvFeature	*PktDrvModel* see 2.1511.3	PktDrv status information.
securityEngineFeature	*SecurityEngineModel* see 2.1511.4	Security Engine status information.
sliceFeature	*SliceModel* see 2.1511.5	Slice status information.
tcamFeature	*TcamModel* see 2.1511.6	TCAM status information.

2.1512 show platform trident security interface INTF detail

see 2.1511.6

2.1513 show platform trident switch-extender counters queue egress

interfaces	*table(Interface name) = SwitchExtenderInterfaceModel* see 2.1513.1	Mapping between interface and queue information.

2.1513.1 SwitchExtenderInterfaceModel

[active]	*boolean*	Connected switchcard active. Only populated for ingress counters.
queues	*table(string) = SwitchExtenderQueueData* see 2.1513.1.1	Mapping between queue name and queue counter information.

2.1513.1.1 SwitchExtenderQueueData

droppedPkts	*integer*	Dropped packet count.

2.1514 show platform trident switch-extender counters queue ingress [...]

interfaces	*table(Interface name) = SwitchExtenderInterfaceBackplaneModel* see 2.1514.1	Mapping between interface and per-switchcard queue information.

2.1514.1 SwitchExtenderInterfaceBackplaneModel

`connections`	*table(string) = SwitchExtenderInterfaceModel* see 2.1513.1	Mapping between connected switchcard and backplane interface.

2.1515 show platform trident tcam cpu-bound [...]

chips	*table(string) = PoliciesModel* see 2.1515.1	Policies, keyed by chip name.

2.1515.1 PoliciesModel

[policies]	*list of PolicyMapModel* see 2.1515.1.1	Policies.

2.1515.1.1 PolicyMapModel

direction	*enum: input | output*	Direction in which policy map is applied.
policyMapName	*string*	Name of the policy map.
policyMapType	*enum: qos | controlPlane*	Type of policy map.
policyRequests	*table(string) = PolicyRequestDetails* see 2.1515.1.1.1	Policy requests, keyed by request name.

2.1515.1.1.1 PolicyRequestDetails

classMaps	*table(string) = ClassMapModel* see 2.1515.1.1.1.2	Class maps, keyed by class map name.
faps	*table(numeric string) = FapSet* see 2.1515.1.1.1.1	FAP set status, keyed by FAP ID.
featureGroupName	*string*	Feature group name.

`featureGroupType`	*enum: fgtInt2Rules* | *fgtSecurityIngressRAclIp6* | *fgtSecurityEgressPAclIp* | *fgtL3CtrlL4MicroBfd* | *fgtSecurityIngressRAclIp* | *fgtIfpDirectflowFull* | *fgtPbrAclIp6* | *fgtIfpOpenflowFullLow* | *fgtIfpDirectflowP4Rt-PostcardIpv6* | *fgtFlowTrackerIp* | *fgtL2CtrlVlanTagValidation* | *fgtL3CtrlMlagCtrlTraffic* | *fgtSecurityIngressPAclMac* | *fgtL3CtrlTtlCheck* | *fgtIfpDirectflowP4Rt-PostcardColor* | *fgtCompressionEtherType* | *fgtVfpDirectflowP4RtVrf* | *fgtIfpOpenflowL2OnlyLow* | *fgtExactMatch* | *fgtSecurityEgressPAclMac* | *fgtIfpOpenflowL3Table* | *fgtIfpDirectflowIpv6Vlan* | *fgtL3CtrlDhcpSnooping* | *fgtIfpOpenflowL2Only* | *fgtPostcardPolicyRules* | *fgtIfpPrefixMroute* | *fgtL2CtrlL3Routing* | *fgtIfpOpenflowIp6ProtoLow* | *fgtL2CtrlLoopProtect* | *fgtSecurityEgressRAclIp6* | *fgtL2CtrlL2Control* | *fgt-SecurityIngressIfpPAclIp6* | *fgtEfpPfcWd* | *fgtFlowTrackerVxlanIp* | *fgtLagHashing* | *fgtPostcardPolicyRulesIpv6* | *fgtIfpOpenflowIp6Proto* | *fgtDirectedBroadcastRules* | *fgtSampTcamClient* | *fgtEgrPolicyQosIp* | *fgtQosEfpEcnCount* | *fgtInvalid* | *fgtCompressionIpProto* | *fgtSecurityEgressPAclIp6* | *fgtFlowTrackerIp6* | *fgtFlowTrackerVxlanIp6* | *fgtFlowTrackerCopyTo-Collector* | *fgtIfpDirectflowP4Rt-PostcardIpv4* | *fgtIntRules* | *fgtSecurityIngressIfpPAcl-Mac* | *fgtCompressionTTL* | *fgtL3CtrlOpenFlow-Management* | *fgtIfpGroupMirrorAcl1* | *fgtTestIfp* | *fgtQosIfpVlanPri* | *fgtL2CtrlStormControl-Management* |	Feature group type.

interfaces	*list of Interface name*	Interfaces on which policy map is applied.
requestConfigType	*enum: deleteConfig* | *updateInterface* | *newConfig* | *invalid* | *updateConfig* | *testDrivenExhaustion* | *revertConfig* | *deleteConfigInRevOrder* | *testDrivenPartial-Exhaustion* | *updateAction* | *updateEntry* | *updateInterfaceInRevOrder*	Last request type.
requestStatusType	*enum: requestFailureUnknown* | *requestFailureOutOfUDF* | *requestFailureOutOfRange-Checker* | *requestSuccess* | *requestNone* | *requestNotApplicable* | *requestFailureIntfProfileId-NotFound* | *requestFailureOutOfSpace* | *requestFailureCounter-ObjectExhaustion* | *requestFailureOutOf-SubportGroupId* | *requestPending* | *requestFailureTimeout* | *requestFailureOutOf-PolicerOrCounter* | *requestFailureUDFGroup-Expansion*	Last known status of this request.
rulesIdsCount	*integer*	Number of rule IDs used by this request.

2.1515.1.1.1.1 FapSet

allPipes	*boolean*	Policy map is applied on all pipes.
pipes	*list of integer*	Pipes on which policy map is applied.

2.1515.1.1.1.2 ClassMapModel

aclName	*string*	Name of ACL matched to class map.
aclRuleIdsCount	*integer*	Number of rule IDs used by ACL.
aclType	*enum: mac* | *ipv6* | *ip*	Type of ACL.
classMapType	*enum: qos* | *controlPlane*	Type of class map.
[policerInfo]	*PolicerInfoModel* see 2.1515.1.1.1.3	Info of Policer attached to class map.
policerType	*enum: PolicerShared* | *PolicerNone* | *PolicerUnshared*	Type of policer.
[ruleIdsDetails]	*table(numeric string) = RuleIdModel* see 2.1515.1.1.1.4	Rule Details, keyed by sequence number.

2.1515.1.1.1.3 PolicerInfoModel

committedBurstSize	*integer*	Comitted burst size.
committedInformationRate	*integer*	Comitted information rate.
mode	*enum: policerModeTrTcm | policerModeSrTcm | policerModeCommitted*	Policer mode.
peakBurstSize	*integer*	Peak burst size.
peakInformationRate	*integer*	Peak information rate.
unit	*enum: policerUnitBytes | policerUnitPackets*	Policer unit.

2.1515.1.1.1.4 RuleIdModel

description	*string*	Description of rule.
hits	*integer*	Number of packets hit this rule.
[policerKey]	*string*	Policer Key.
[qualifiers]	*table(string) = Qualifier* see 2.1515.1.1.1.5	Qualifiers, keyed by qualifier type.
[rangeCheckers]	*table(string) = RangeChecker* see 2.1515.1.1.1.6	Range Checkers, keyed by range checker type.
ruleEnabled	*boolean*	Rule is enabled in hardware.

2.1515.1.1.1.5 Qualifier

data	*string*	Qualifier data.
mask	*string*	Qualifier mask.

2.1515.1.1.1.6 RangeChecker

high	*integer*	Highest range.
low	*integer*	Lowest range.

2.1516 show platform trident tcam profile [...]

chips	*table(string) = TcamChipInfo* see 2.1516.1	A mapping from chip to it's groups.

2.1516.1 TcamChipInfo

groups	*table(numeric string) = ChipGroupInfo* see 2.1516.1.1	A mapping from group number to its qualifiers.

2.1516.1.1 ChipGroupInfo

currentQualifiers	*list of string*	Current Qualifiers.
dontCareQualifiers	*list of string*	Qualifier defaulting to Dont-Care (diff from config).
[dontCareQualifiersEntry]	*table(numeric string) = PerIdInfo* see 2.1516.1.1.1	Mapping from Entry ID to Qualifiers defaulting to Dont-Care.
[reservedSlices] **	*list of integer*	The TCAM slices allocated to this group. **Since: 4.30.0

2.1516.1.1.1 PerIdInfo

qualifiers	*list of string*	Qualifiers associated with this ID.

2.1517 show platform trident tcam qos [...]

see 2.1515

2.1518 show platform trident tcam status [...]

features	*table(string) = AggregatorFeatureInfo* see 2.1518.1	A mapping of feature name to aggregate state, keyed by feature name.

2.1518.1 AggregatorFeatureInfo

attributes	*table(string) = AggregatorFeatureAttributeInfo* see 2.1518.1.1	A mapping of attribute name to values, keyed by attribute name.
requests	*table(string) = AggregatorRequestInfo* see 2.1518.1.2	A mapping of request name to feature requests, keyed by request name.

2.1518.1.1 AggregatorFeatureAttributeInfo

globalFeatureConfigWarm	*boolean*	Global feature config.
globalFeatureStatusWarm	*boolean*	Global feature status.
slicesFeatureStatus	*table(string) = boolean*	A mapping of slice name to feature status, keyed by slice name.

2.1518.1.2 AggregatorRequestInfo

genId	*integer*	Reqeust config genId.
globalRequestStatus	*enum: requestFailureUnknown | requestFailureOutOfRangeChecker | requestSuccess | requestNone | requestNotApplicable | requestFailureIntfProfileIdNotFound | requestFailureOutOfSpace | requestPending | requestFailureTimeout*	Global request status.
requestConfigType	*enum: deleteConfig | updateInterface | newConfig | invalid | updateConfig | revertConfig | updateAction*	Request config type.
slicesRequestStatus	*table(string) = string*	A mapping of slice name to request status, keyed by slice name.

2.1519 show platform trident tcam summary [...]

switches **	*table(string) = StrataTcamSummaryChip* see 2.1519.1	Mapping from switches' name to their TCAM summary. **Since: 4.29.2

2.1519.1 StrataTcamSummaryChip

features	*table(string) = StrataTcamSummaryFeatureModel* see 2.1519.1.1	Mapping from features to slice groups they occupy.

2.1519.1.1 StrataTcamSummaryFeatureModel

bankType	*enum: flowtracker | efp | ifp | vfp*	Stage of TCAM slices.
banks	*list of integer*	Index of TCAM slices.

2.1520 show platform trident vfi mc-index

l3VfiMcIndexInfo	*table(numeric string) = L3VfiMcIndexInfo* see 2.1520.1	VFI MC Index table indexed by VLAN Id.

2.1520.1 L3VfiMcIndexInfo

[routedMc]	*integer*	Routed MC Index allocated for VFI.
[umcCore]	*integer*	UMC Index including core and edge ports.
[umcEdge]	*integer*	UMC Index including edge ports only.
uucCore	*integer*	UUC Index including core and edge ports.

uucEdge	*integer*	UUC Index including edge ports only.

2.1521 show platform trident vfi virtual-ports

revision 2

PhysicalInterfaceVps	*table(string) = L3VfiPhysicalInterfaceVp* see 2.1521.1	VFI virtual port table indexed by the physical port.
SystemInterfaceVps	*table(string) = L3VfiSystemInterfaceVp* see 2.1521.2	VFI virtual port table indexed by the system interface.

2.1521.1 L3VfiPhysicalInterfaceVp

vportAttributes	*list of L3VfiPhysicalInterfaceVpAttributes* see 2.1521.1.1	VFI attributes associated with the interface.

2.1521.1.1 L3VfiPhysicalInterfaceVpAttributes

dvp	*integer*	Destination virtual port associated with the interface.
egrVlanFilter **	*boolean*	Flag to indicate VLAN membership check on packets egressing the interface. **Since: 4.22.0
sdtagNextHopId	*integer*	SDTAG nexthop identifier for the interface.
tagged	*boolean*	Flag to indicate whether packets going out of the interface have to be tagged for this VLAN.
[vlanId]	*integer*	VLAN identifier.

2.1521.2 L3VfiSystemInterfaceVp

vportAttributes	*list of L3VfiSystemInterfaceVpAttributes* see 2.1521.2.1	Attributes associated with the Virtual Port.

2.1521.2.1 L3VfiSystemInterfaceVpAttributes

dvp	*integer*	Virtual port identifier.

2.1522 show platform trident vfi vlan map

pendingVlans **	*list of integer*	Pending VLANs due to resource exhaustion. **Since: 4.26.2
vfis	*table(numeric string) = integer*	A mapping between VFI index and VLAN.

2.1523 show platform trident vlan membership egr-vp

vpVlanList	*list of VpVlanMembershipEntry* see 2.1523.1	VLAN membership table list.

2.1523.1 VpVlanMembershipEntry

[entryId]	*integer*	hardware entry ID.
interface	*interface name*	Front Panel Interface.
logicalPort	*integer*	Switchcard Logical Port number.
modId	*integer*	Switchcard ModId.
stpState	*string*	Spanning Tree state.
[untag]	*string*	untag egress packet.
vlanId	*integer*	VLAN ID.

2.1524 show platform trident vlan membership egr-vp unprogrammed

vpVlanList	*list of VpVlanMembershipEntry* see 2.1523.1	VLAN membership table list.

2.1525 show platform trident vlan membership ing-vp

vpVlanList	*list of VpVlanMembershipEntry* see 2.1523.1	VLAN membership table list.

2.1526 show platform trident vlan membership ing-vp unprogrammed

vpVlanList	*list of VpVlanMembershipEntry* see 2.1523.1	VLAN membership table list.

2.1527 show platform trident vlan profile

vlanProfiles	*table(numeric string) = VlanProfileEntry* see 2.1527.1	VLAN profiles keyed by VLAN profile ID.

2.1527.1 VlanProfileEntry

broadcastMask	*enum: a | ab | none | b*	BLOCK_MASK_A and BLOCK_MASK_B applied to broadcast packets.
knownMcastMask	*enum: a | ab | none | b*	BLOCK_MASK_A and BLOCK_MASK_B applied to known multicast packets.
l2Ipv4Pfm	*enum: all | none | unknown*	Mcast flood mode for L2.

l3Ipv4Pfm	*enum: all | none | unknown*	Mcast flood mode for IPv4.
l3Ipv6Pfm	*enum: all | none | unknown*	Mcast flood mode for IPv6.
unknownMcastMask	*enum: a | ab | none | b*	BLOCK_MASK_A and BLOCK_MASK_B applied to unknown multicast packets.
unknownUcastMask	*enum: a | ab | none | b*	BLOCK_MASK_A and BLOCK_MASK_B applied to unknown unicast packets.
vlanIP4McastL2Disabled	*boolean*	L2-only forwarding of IPMCv4 is disabled.
vlanIP6McastL2Disabled	*boolean*	L2-only forwarding of IPMCv6 is disabled.

2.1528 show platform trident vlan setting

vlans	*table(numeric string) = VlanSettingEntry* see 2.1528.1	VLAN settings keyed by VLAN ID.

2.1528.1 VlanSettingEntry

egressValid	*boolean*	Egress valid for VLAN.
forwardingVlan	*integer*	VLAN ID forwarding to.
ingressValid	*boolean*	Ingress valid for VLAN.
profileId	*integer*	VLAN's profile ID.
stg	*integer*	Spanning tree group ID.
vlanClassId	*integer*	ID of VLAN class.

2.1529 show platform trident vlan status

vlans	*table(numeric string) = VlanStatusEntry* see 2.1529.1	VLAN Statuses Keyed by VLAN ID.

2.1529.1 VlanStatusEntry

interfaces	*list of string*	List of interfaces of one VLAN ID.

2.1530 show platform trident vlan stg

topologies	*table(numeric string) = VlanStgEntry* see 2.1530.1	Spanning tree information of each topology keyed by topology ID.

2.1530.1 VlanStgEntry

[logicalFapIds]	*table(numeric string) = VlanStgLogicalFapEntry* see 2.1530.1.1	Interfaces for logical FAP keyed by logical FAP ID.
topoName	*string*	Name of topology.
vlans	*list of integer*	List of VLAN IDs in the spanning tree.

2.1530.1.1 VlanStgLogicalFapEntry

[interfaces]	*table(string) = VlanStgInterfaceEntry* see 2.1530.1.1.1	Interfaces in the spanning tree keyed by interface name.

2.1530.1.1.1 VlanStgInterfaceEntry

[modId]	*integer*	Switchcard module ID.
[portId]	*integer*	Switchcard port ID.
stpState	*enum: learning | discarding | unknown | forwarding*	Spanning tree state.

2.1531 show platform trident vxlan local-vtep

interfaces	*table(Interface name) = VxlanLocalVtepV4V6InterfaceModel* see 2.1531.1	VXLAN interfaces.

2.1531.1 VxlanLocalVtepV4V6InterfaceModel

[vtepV4]	*VxlanLocalVtepInterfaceInfoModel* see 2.1531.1.1	IPv4 VTEP.
[vtepV6]	*VxlanLocalVtepInterfaceInfoModel* see 2.1531.1.1	IPv6 VTEP.

2.1531.1.1 VxlanLocalVtepInterfaceInfoModel

localVtep	*IP address*	Local VTEP address.
ttl	*integer*	Time to live (hops).
udpPort	*integer*	UDP port.

2.1532 show platform trident vxlan multicast route underlay [...]

underlayMroutes	*list of MrouteKeyToEMEntryModel* see 2.1532.1	A List of Vxlan Underlay Mroutes.

2.1532.1 MrouteKeyToEMEntryModel

groupIp	*IP address*	Group IP address.
interfaceName	*interface name*	Interface name.
isActive	*boolean*	Underlay Mroute is active.
isProgrammed	*boolean*	Underlay Mroute is Programmed.
sourceIp	*IP address*	Source IP address.
vlanId	*integer*	VLAN Identifier.

2.1533 show platform trident vxlan multicast tunnel [...]

tunnels	*table(numeric string) = McastTunnelEntryModel* see 2.1533.1	A mapping of Encap Id to VXLAN Multicast Tunnels.

2.1533.1 McastTunnelEntryModel

dvpIndex	*integer*	Hardware destination index.
[groups]	*list of integer*	List of groups.
portNextHops	*list of PortNextHopModel* see 2.1533.1.1	List of Port-NextHops.
stale	*boolean*	Tunnel information is stale.
tunnelIp	*IP address*	VXLAN Multicast tunnel IP address.

2.1533.1.1 PortNextHopModel

nextHopId	*integer*	NextHop index.
port	*interface name*	Core interface.

2.1534 show platform trident vxlan multihoming groups non-peering

filteredGroups	*list of FilteredGroupRow* see 2.1534.1	Non-peering groups.

2.1534.1 FilteredGroupRow

coreGroup	*integer*	Core Group ID.
edgeGroup	*integer*	Edge Group ID.
filteredGroup	*integer*	Filtered Group ID.
vlan	*integer*	VLAN ID.

2.1535 show platform trident vxlan multihoming groups peering

peeringGroups	*list of PeeringGroupRow* see 2.1535.1	Peering groups.

2.1535.1 PeeringGroupRow

group	*integer*	Group ID.
tcamEntryId	*integer*	TCAM Entry ID.
vni	*integer*	VXLAN VNI.
vp	*integer*	Virtual Port.

2.1536 show platform trident vxlan next-hop

[herMlagVteps]	*table(string) = integer*	L3Mc Next-Hops, keyed by Remote MLAG HER VTEPs.
[herVteps]	*table(string) = HerVtepNextHopsModel* see 2.1536.1	L3Mc Next-Hops, keyed by HER VTEPs.
[localVtepNextHops]	*integer*	Total local remote VTEP Next-Hops.
[memberTunnelVteps]	*table(string) = integer*	L3Mc Next-Hops, keyed by Member tunnel VTEPs.
[mlagVtepNextHops]	*integer*	Total Mlag VTEP Next-Hops.
[remoteVteps]	*table(string) = integer*	Next-Hops, keyed by Remote VTEPs.
[vfis]	*table(string) = VfiNextHopsModel* see 2.1536.2	Next-Hops, keyed by VFI id.

2.1536.1 HerVtepNextHopsModel

[dropNextHop]	*boolean*	Next-Hops not part of L3MC Next-Hops collection.
[nextHopIds]	*list of integer*	Next-Hops in L3MC Next-Hops collection.

2.1536.2 VfiNextHopsModel

[mcIndex]	*integer*	Multicast index of VLAN.
[ports]	*table(Interface name) = PortVlanModel* see 2.1536.2.1	Next-Hops, keyed by L3 port.

2.1536.2.1 PortVlanModel

[multiVtepMlagNextHops]	*list of integer*	Next-hops for multiVTEP MLAG.
[sdtagNextHopId]	*integer*	Sdtag Next-Hop id.
[singleVtepMlagNextHops]	*list of integer*	Next-hops for singleVTEP MLAG.

2.1537 show platform trident vxlan port-to-next-hop

vxlanPortToNhTableColl	*table(Interface name) = VxlanPortToNhTable* see 2.1537.1	Port-To-Next-Hop Table for all Ports.

2.1537.1 VxlanPortToNhTable

installedNhId	*integer*	Next Hop ID Installed in Port-To-Next-Hop Table.
portToNextHops	*list of VxlanPortToNhEntryNhAttr* see 2.1537.1.1	Next Hop Entry Attributes.
tableOverflow	*boolean*	Flag indicating whether Port-To-Next-Hop Table is currently in Resource Overflow state.

2.1537.1.1 VxlanPortToNhEntryNhAttr

[installError]	*boolean*	Error installing this Entry in H/W?
nextHopId	*integer*	Port To Next Hop Table Index.
nextHopMac	*MAC address*	Next Hop MAC Address.

2.1538 show platform trident vxlan port-to-next-hop overflow

see 2.1537

2.1539 show platform trident vxlan routing-info

[recircChannel]	*interface name*	Recirculation Interface.
routerMac	*MAC address*	Router MAC Address.
[varpVtepIp]	*IP address*	VARP VTEP IP address.
[varpVtepSvp]	*integer*	VARP VTEP SVP.
[virtualRouterMac]	*MAC address*	Virtual Router MAC Address.
vlans	*table(numeric string) = VlanToSvpEntry* see 2.1539.1	A mapping from VLAN to SVP info.

2.1539.1 VlanToSvpEntry

routerMacConfigured	*boolean*	Router MAC is configured.
[routerMacNotInShadow]	*boolean*	Router MAC not in MAC shadow.
[routerMacSetToDiscard]	*boolean*	Router MAC is set to discard.
svp	*integer*	SVP.
[virtualRouterMac-Configured]	*boolean*	Virtual Router MAC configured.
[virtualRouterMacNotIn-Shadow]	*boolean*	Virtual Router MAC present in shadow.
[virtualRouterMacSetTo-Discard]	*boolean*	Virtual Router MAC set to discard.
vlanXlateEntryPresent	*boolean*	VLAN XLATE Entry is present.

2.1540 show platform trident vxlan vlan [...]

vlans	*table(numeric string) = VxlanVlanEntry* see 2.1540.1	Mapping from VLAN to VXLAN VLAN mapping.

2.1540.1 VxlanVlanEntry

[mcastIndex]	*integer*	Mcast Index.
vfi	*integer*	VFI.
[virtualPorts]	*table(Interface name) = integer*	Mapping from Virtual Ports to SVP. Populated only for detail version of command.
[vlanSvp]	*integer*	VLAN SVP.
vni	*integer*	VNI.

2.1541 show platform trident vxlan vlan summary

[recircChannel]	*interface name*	Recirculation interface.
[varpSvp]	*integer*	Varp Vtep SVP allocated.
[varpVtepIp]	*IP address*	VARP VTEP IP address.
[virtualRouterMac]	*MAC address*	Vxlan Virtual Router MAC Address.
[vxlanRouterMac]	*MAC address*	Vxlan Router MAC Address.
vxlanVlansWithVirtualPorts	*integer*	Vxlan VLANs with Virtual Ports.
vxlanVlansWithVniMapping	*integer*	Vxlan VLANs with VNI Mappings.

2.1542 show platform trident vxlan vp-usage

allConfigCount	*integer*	Virtual port usage presumed.
allStatusCount	*integer*	Virtual port usage actual.
edgeVpCount	*integer*	Edge virtual port count.
reservedCount	*integer*	Reserved virtual port count.
routableVlanCount	*integer*	Routable VLAN count.
totalUsedCount	*integer*	Total used virtual port count.
vpAllocFailures	*integer*	Virtual port allocation failures.
vpAllocRequests	*integer*	Virtual port allocation requests.
vpFreeRequests	*integer*	Virtual port free requests.
vpTableSize	*integer*	Virtual port table size.
vtepCount	*integer*	Virtual tunnel endpoint count.
vtepGroupCount	*integer*	Virtual tunnel endpoint group count.

2.1543 show platform trident vxlan vtep [...]

[herTunnelVteps]	*table(numeric string) = VxlanRemoteVtepInfoModel* see 2.1543.1	VXLAN HER tunnel VTEPs, keyed by remote ID.
[memberTunnelVteps]	*table(numeric string) = VxlanRemoteVtepInfoModel* see 2.1543.1	VXLAN member tunnel VTEPs, keyed remote ID.

[vxlanTunnelVteps]	*table(numeric string) = VxlanRemoteVtepInfoModel* see 2.1543.1	VXLAN tunnel VTEPs, keyed by remote ID.

2.1543.1 VxlanRemoteVtepInfoModel

[nhInfo]	*VxlanRemoteVtepNextHop-InfoModel* see 2.1543.1.1	VXLAN remote VTEP next hop information.
remoteVtep	*IP address*	VXLAN tunnel IP address.

2.1543.1.1 VxlanRemoteVtepNextHopInfoModel

ecmp	*boolean*	VTEP has ECMP next hops.
nextHopId	*integer*	ECMP next hop ID.
rtNhUpdates	*integer*	Number of route next hop updates.

2.1544 show poe [...]

detail **	*boolean*	Show detailed status of each 2-pairs. **Since: 4.29.2
poePorts	*table(string) = PoePort* see 2.1544.1	PoE ports.

2.1544.1 PoePort

[alternative] **	*PoeChannel* see 2.1544.1.1	Details on alternative channel. **Since: 4.29.2
[current]	*floating point number*	Output current (milliamps).
[grantedPower] **	*floating point number*	PoE port granted power (watts). **Since: 4.22.0
[legacyDetectEnabled] **	*boolean*	Legacy detect enabled. **Since: 4.22.0
[lldpEnabled]	*boolean*	LLDP enabled.
[misconfigured] **	*boolean*	Legacy detect and port mode misconfiguraton. **Since: 4.24.2
[pdClass]	*string*	Powered device class.
[portCapacity] **	*floating point number*	PoE port capacity (watts). **Since: 4.23.1
portName	*string*	PoE port name.
[portPairset] **	*enum: altA | fourPair | altB*	Enabled pairset(s). **Since: 4.24.2
portPresent	*boolean*	PoE port present.
[portPriority] **	*enum: high | critical | medium | low*	PoE port priority. Available power is given to higher priority ports. A power supply failure can reduce the available power. **Since: 4.22.1
[portState]	*enum: unknown | fault | adminShutdown | invalid | linkDown | powered | disabled | detecting | classified | mismatch | overloaded | failed*	PoE port state.
[power]	*floating point number*	Output power (watts).
[powerLimit]	*floating point number*	PoE port power limit (watts).
[primary] **	*PoeChannel* see 2.1544.1.1	Details on primary channel. **Since: 4.29.2
[pseEnabled]	*boolean*	PSE functionalities enabled.
[temperature]	*floating point number*	Temperature (celsius).
[voltage]	*floating point number*	Output voltage (volts).

2.1544.1.1 PoeChannel

[current]	*floating point number*	Output current (milliamps).
[pairSet]	*string*	Pairset of the channel.
[pdClass]	*string*	Powered device class.

[portState]	*enum: unknown	fault	adminShutdown	invalid	linkDown	powered	disabled	detecting	classified	mismatch	overloaded	failed*	PoE channel state.
[power]	*floating point number*	Output power (watts).
[pseLimit]	*floating point number*	PoE port power limit (watts).
[voltage]	*floating point number*	Output voltage (volts).

2.1545 show poe neighbor [...]

poePorts	*table(Interface name) = PoePortNbrInfo* see 2.1545.1	A mapping of PoE port to its information.
powerBudget	*floating point number*	Total power budget in watts.
remainingPower	*floating point number*	Remaining power in watts.
usedPoePower	*floating point number*	Total used PoE output power in watts.

2.1545.1 PoePortNbrInfo

[lldpSystemDesc]	*list of string*	Neighbor system description.
[portState]	*enum: unknown	fault	adminShutdown	invalid	linkDown	powered	disabled	detecting	classified	mismatch	overloaded	failed*	PoE port state.
[power]	*floating point number*	Output power (watts).

2.1546 show policing [...]

revision 2

pktSizeAdjProfiles	*table(string) = PolicerPktSizeAdjProfileModel* see 2.1546.2	Policer packet size adjustment profiles index by their name.
policerInstances	*table(string) = PolicerInstanceModel* see 2.1546.1	Policer instances indexed by their name.
profiles	*table(string) = ProfileModel* see 2.1546.3	Profiles indexed by their name.

2.1546.1 PolicerInstanceModel

[configuredIngressIntfs]	*table(Interface name) = boolean*	A mapping of interface to whether it is configured on ingress.
policerName	*string*	Policer instance name.
profileName	*string*	Profile associated with Policer instance.

2.1546.2 PolicerPktSizeAdjProfileModel

[configuredIngressIntfs]	*table(Interface name) = HwProgrammingEnumModel* see 2.1546.2.1	A mapping of interface to whether it is configured on ingress.
pktSizeAdj	*integer*	Packet size adjustment configuration associated with profiles.
profileName	*string*	Profile name.

2.1546.2.1 HwProgrammingEnumModel

hwState	*enum: partial | inactive | active*	Hardware programming state.

2.1546.3 ProfileModel

[configuredIngressIntfs]	*table(Interface name) = boolean*	A mapping of interface to whether it is configured on ingress.
police	*PoliceModel* see 2.1546.3.1	Policing configuration associated with profiles.
[policerAssociations]	*list of string*	Policers to which the profile is attached.
profileName	*string*	Profile name.

2.1546.3.1 PoliceModel

bc	*integer*	Committed Burst Rate.
bcUnit	*enum: mbytes | bytes | kbytes | packets*	burst unit.
be	*integer*	Excess Burst Size.

beUnit	*enum: mbytes | bytes | kbytes | packets*	burst unit.
cir	*integer*	Committed Information Rate.
cirUnit	*enum: kbps | bps | mbps | pps*	CIR unit.
[counters]	*table(string) = PoliceCounters* see 2.1546.3.1.1	A mapping from direction ('input' or 'output') to per interface police counters.
[greenCounter]	*boolean*	Green counter is supported.
[perSviRedCounterWithSharedPolicer]	*boolean*	Red counter per interface in shared policer mode is supported.
pir	*integer*	Peak Information Rate.
pirUnit	*enum: kbps | bps | mbps | pps*	PIR unit.
[policerName]	*string*	Policer name.
[redCounter]	*boolean*	Red counter is supported.
[yellowActions]	*PolicerActions* see 2.1546.3.1.2	Yellow Actions applicable only for trTCM Policer.

2.1546.3.1.1 PoliceCounters

[interfaces]	*table(string) = IntfPoliceCounters* see 2.1546.3.1.1.1	Police Counters.

2.1546.3.1.1.1 IntfPoliceCounters

[conformedBitsRate] **	*floating point number*	Conformed bits rate in bps. **Since: 4.29.2
conformedBytes	*integer*	Conformed number of bytes to policer.
conformedPackets	*integer*	Conformed number of packets to policer.
[droppedBitsRate] **	*floating point number*	Dropped bits rate in bps. **Since: 4.29.2
[exceededBitsRate] **	*floating point number*	Excess bits rate in bps. **Since: 4.29.2
exceededBytes	*integer*	Exceeded number of bytes to policer.
exceededPackets	*integer*	Exceeded number of packets to policer.
[updateInterval] **	*floating point number*	Stats update interval in seconds. **Since: 4.29.2
yellowBytes **	*integer*	Conformed number of bytes to PIR and not CIR. **Since: 4.20.5
yellowPackets **	*integer*	Conformed number of packets to PIR and not CIR. **Since: 4.20.5

2.1546.3.1.2 PolicerActions

[dropPrecedence]	*boolean*	Mark Drop-Prcedence.
[dscp]	*integer*	DSCP value.
[dscpConfiguredAsName]	*boolean*	The DSCP value was configured as a name.

2.1547 show policing interfaces [...]

intfCpuPolicerInfo **	*table(Interface name) = InterfacePolicerModel* see 2.1547.1	CPU interface-policer info keyed by interface. **Since: 4.29.0
intfEgrPolicerInfo **	*table(Interface name) = InterfacePolicerModel* see 2.1547.1	Egress interface-policer info keyed by interface. **Since: 4.29.0
intfPolicerInfo	*table(Interface name) = InterfacePolicerModel* see 2.1547.1	Ingress interface-policer info keyed by interface.

2.1547.1 InterfacePolicerModel

isConfigured	*HwProgrammingEnumModel* see 2.1546.2.1	Interface programming status.
[policerName]	*string*	Group policer.
[portChannelMembers]	*table(Interface name) = HwProgrammingEnumModel* see 2.1546.2.1	A mapping of member interface to whether it is configured.
[profileName]	*string*	Profile name.

2.1548 show policing interfaces counters [...]

cpuInterfaces **	*table(Interface name) = InterfacePolicingCounterModel* see 2.1548.1	Interface CPU policing counter info keyed by interface. **Since: 4.29.0
egrInterfaces **	*table(Interface name) = InterfacePolicingCounterModel* see 2.1548.1	Interface egress policing counter info keyed by interface. **Since: 4.29.0
interfaces	*table(Interface name) = InterfacePolicingCounterModel* see 2.1548.1	Interface ingress policing counter info keyed by interface.

2.1548.1 InterfacePolicingCounterModel

counters	*IntfPoliceCounters* see 2.1546.3.1.1.1	Policer Counters.
mode	*enum: trtcm | committed*	Policer mode.
[policerName]	*string*	Group policer.
[profileName]	*string*	Profile name.

2.1549 show policy-map (copp | (interface control-plane)) PMAP

`intfIds`	*list of Interface name*	List of associated interfaces.

2.1550 show policy-map (copp | (interface control-plane)) copp-system-policy

revision 4

mapType	*enum: mapPdp | mapControlPlane | mapCoppCli | mapQos*	Determines if it is a CoPP or Qos or PDP policy map.
[maxCmapsPerPolicy]	*integer*	Maximum number of class maps supported per policy map.
policyMaps	*table(string) = PolicyMapModel* see 2.1550.1	List of all policy maps.

2.1550.1 PolicyMapModel

[activeIntfs]	*table(string) = InterfaceList* see 2.1550.1.3	Dictionary of list of interfaces with the policy active for the directions('input', 'output').
classMaps	*table(string) = ClassMap* see 2.1550.1.1	List of class maps for a given policy map.
[configuredIntfs]	*table(string) = InterfaceList* see 2.1550.1.3	Dictionary of list of interfaces with the policy configured for the directions('input', 'output').
hwStatus	*table(string) = HwStatus* see 2.1550.1.2	Hardware status for direction ('input', 'output').
[inactiveIntfs]	*table(string) = InterfaceList* see 2.1550.1.3	Dictionary of list of interfaces with the policy inactive for the directions('input', 'output').
mapType	*enum: mapPdp | mapControlPlane | mapQos*	Determines if it is a CoPP or Qos or PDP policy map.
name	*string*	Name of policy map.
[shared]	*boolean*	True if the policy map is shared.
spApplied	*table(string) = boolean*	True for the directions('input', 'output')having service policy.
summary	*boolean*	Print summary info.

2.1550.1.1 ClassMap

[bandwidth]	*BandwidthModel* see 2.1550.1.1.1	Bandwidth associated with the policy class, applicable only for CoPP policy maps.
[classMapStatistics]	*table(string) = ClassMapStatistics* see 2.1550.1.1.3	A mapping from direction ('input' or 'output') to class maps counter statistics.
classPrio	*integer*	Class priority - lower value has precedence.
[coppActionPoliceSupported]	*boolean*	Control plane policer supported.
[count]	*boolean*	Count number of packets matching the policy class.

[intfPacketCounters]	*table(string) = IntfPacketCounters* see 2.1550.1.1.4	A mapping from direction ('input' or 'output') to per interface packet counters.
[mapType]	*enum: mapPdp | mapControlPlane | mapQos*	Determines if the class map is for CoPP or Qos or PDP policy map.
[match]	*table(string) = Match* see 2.1550.1.1.5	Matching access groups.
[matchCondition]	*enum: unspecified | matchConditionAny*	Match condition.
[matchedPackets]	*table(string) = ClassMapMatchPacketCounterCollection* see 2.1550.1.1.2	A mapping from direction ('input' or 'output') to number of packets matched.
name	*string*	Name of class map.
[police]	*PoliceModel* see 2.1546.3.1	Ingress policing associated with the policy class.
[qosParams]	*QosParams* see 2.1550.1.1.6	Value of Qos parameters, applicable only for Qos policy maps.
[shape]	*ShapeRateModel* see 2.477.1.1.1.2	Shape rate associated with the policy class, applicable only for CoPP policy maps.

2.1550.1.1.1 BandwidthModel

rate	*integer*	Bandwith supported.
unit	*enum: kbps | unspecified | gbps | pps*	Bandwidth unit.

2.1550.1.1.2 ClassMapMatchPacketCounterCollection

[interfaces]	*table(string) = ClassMapMatchPacketCounter* see 2.1550.1.1.2.1	Interface Matched Packets.

2.1550.1.1.2.1 ClassMapMatchPacketCounter

[packets]	*integer*	Number of packets matched.

2.1550.1.1.3 ClassMapStatistics

[conformedBitsRate]	*floating point number*	Conformed bits rate in bps.
[droppedBitsRate]	*floating point number*	Dropped bits rate in bps.
[exceededBitsRate]	*floating point number*	Excess bits rate in bps.
[updateInterval]	*floating point number*	Stats update interval in seconds.

2.1550.1.1.4 IntfPacketCounters

[dropPackets]	*integer*	Number of packets dropped.
[outPackets]	*integer*	Number of out packets.

2.1550.1.1.5 Match

[dscpValue]	*integer*	DSCP equivalent value in digits.
[l2ParamsValue]	*MatchL2Params* see 2.239.1.1.1	L2 Params values configured.
[matchDscpNameValid]	*boolean*	Flag to identify if string value of DSCP is used or numeric value.
[matchEcnValue]	*enum: non-ect* | *ce* | *ect* | *dontCare* | *ect-ce*	ECN value configured.
matchRule	*enum: matchIpAccessGroup* | *matchDscpEcn* | *matchMacAccessGroup* | *matchL2Params* | *matchMplsTrafficClass* | *matchIpv6AccessGroup*	Matching rule for the type.

2.1550.1.1.6 QosParams

[cos]	*integer*	COS value.
[drop]	*boolean*	Drop Packets.
[dropActionRestricted]	*boolean*	Is drop action restricted on the system.
[dropPrecedence]	*integer*	Drop Precedence value.
[dscp]	*integer*	DSCP value.
[dscpConfiguredAsName]	*boolean*	Flag to identify if the DSCP value was configured as a name.
[trafficClass]	*integer*	Traffic Class value.

2.1550.1.2 HwStatus

status	*enum: hwPrgmStatusSuccess* | *hwPrgmStatusFail* | *hwPrgmStatusInProgress*	Hardware programming status.
[unitsProgrammed]	*integer*	Number of units programmed.

2.1550.1.3 InterfaceList

[interfaces]	*list of Interface name*	List of interfaces.

2.1551 show policy-map PMAP class CMAP [...]

revision 4

[classMap]	*ClassMap* see 2.1550.1.1	Wrapper around a class map.

2.1552 show policy-map PMAP counters [...]

revision 4

see 2.1550

2.1553 show policy-map [...]

revision 4

see 2.1550

2.1554 show policy-map counters [...]

revision 4

see 2.1550

2.1555 show policy-map interface (INTF | SVI_INTF | VTI_INTF) [...]

revision 4

see 2.1550

2.1556 show policy-map interface INTF type control-plane

revision 4

see 2.1550

2.1557 show policy-map type (copp | control-plane) (copp-system-policy | PMAP) class CMAP

revision 4

see 2.1551

2.1558 show policy-map type (copp | control-plane) [...]

revision 4

see 2.1550

2.1559 show policy-map type (copp | control-plane) counters [...]

revision 4

see 2.1550

2.1560 show policy-map type pbr [...]

revision 2

intf	*table(string) = string*	Policy map to intf map.
mapType	*enum: mapTrafficPolicy | mapSamplePolicy | mapServerLoadBalancer | mapQos | mapTapAgg | mapPbr | mapFlowspec*	Map type.
policyMaps	*table(string) = PolicyMapModel* see 2.1560.1	List of policy maps.

2.1560.1 PolicyMapModel

[appliedIngressIntfs]	*list of string*	List of interfaces with the policy applied on ingress.
[classMap]	*table(numeric string) = PolicyClassMapModel* see 2.1560.1.1	Mapping of priority to class map status.
[configuredIngressIntfs]	*list of string*	List of interfaces with the policy configured on ingress.
[configuredIngressIntfsAsFallback]	*list of string*	List of interfaces with the policy configured on ingress as fallback policy.
mapType	*enum: mapTrafficPolicy | mapSamplePolicy | mapServerLoadBalancer | mapQos | mapTapAgg | mapPbr | mapFlowspec*	Map type.
name	*string*	Name of policy map.
[rawClassMap]	*table(string) = ClassMapModel* see 2.241.1	Pseudo class maps for single match statements.

2.1560.1.1 PolicyClassMapModel

Could also be an instance of:
PbrPolicyClassMapModel 2.1560.1.2

[actionSetConfig] **	*ActionSetModel* see 2.1560.1.1.18	Action set configuration. **Since: 4.29.1
classMap	*ClassMapModel* see 2.241.1	Class map configuration.
[configuredAction]	*table(string) = ActionModel* see 2.1560.1.1.1	List of class actions.
name	*string*	Class map name.

2.1560.1.1.1 ActionModel

Could also be an instance of:
DropActionModel 2.1560.1.1.2
FwdActionModel 2.1560.1.1.3

NexthopGroupModel 2.1560.1.1.4
NexthopModel 2.1560.1.1.5
NoActionModel 2.1560.1.1.6
SetDscpModel 2.1560.1.1.7
SetTtlModel 2.1560.1.1.8
TapAggActionSetModel 2.1560.1.1.9
TapAggDropActionModel 2.1560.1.1.10
TapAggGroupModel 2.1560.1.1.11
TapAggHeaderRemoveActionModel 2.1560.1.1.12
TapAggIdTagModel 2.1560.1.1.13
TapAggMacAddressActionModel 2.1560.1.1.14
TapAggNexthopGroupModel 2.1560.1.1.15
TapAggStripHdrModel 2.1560.1.1.16
TapAggTimestampHeaderModel 2.1560.1.1.17

`actionType`	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.2 DropActionModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.3 FwdActionModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.4 NexthopGroupModel

actionType	*enum: setDscp | setTtl | sample | setHeaderRemove | sflow | setTc | setMacAddress | actionTest2 | loadBalance | actionTest1 | sampleAll | setNexthopGroup | setVrf | deny | setIdentityTag | actionSet | permit | actionTest3 | actionGoto | mirror | setAggregationGroup | useVrfSecondary | stripHeaderBytes | count | accessList | setTimestampHeader | reportFlow | maxActionEnum | setVrfSecondary | police | log | setNexthop | actionShaping | fwdAction*	class action type.
nexthopGroup	*string*	Next hop group name.

2.1560.1.1.5 NexthopModel

actionType	*enum: setDscp | setTtl | sample | setHeaderRemove | sflow | setTc | setMacAddress | actionTest2 | loadBalance | actionTest1 | sampleAll | setNexthopGroup | setVrf | deny | setIdentityTag | actionSet | permit | actionTest3 | actionGoto | mirror | setAggregationGroup | useVrfSecondary | stripHeaderBytes | count | accessList | setTimestampHeader | reportFlow | maxActionEnum | setVrfSecondary | police | log | setNexthop | actionShaping | fwdAction*	class action type.
nexthops	*table(numeric string) = IpGenericAddr* see 2.296.1.1.1.1	Set of next hops.
recursive	*boolean*	recursive or not.
vrfName	*string*	VRF name.

2.1560.1.1.6 NoActionModel

actionType	*enum: setDscp | setTtl | sample | setHeaderRemove | sflow | setTc | setMacAddress | actionTest2 | loadBalance | actionTest1 | sampleAll | setNexthopGroup | setVrf | deny | setIdentityTag | actionSet | permit | actionTest3 | actionGoto | mirror | setAggregationGroup | useVrfSecondary | stripHeaderBytes | count | accessList | setTimestampHeader | reportFlow | maxActionEnum | setVrfSecondary | police | log | setNexthop | actionShaping | fwdAction*	class action type.

2.1560.1.1.7 SetDscpModel

actionType	*enum: setDscp | setTtl | sample | setHeaderRemove | sflow | setTc | setMacAddress | actionTest2 | loadBalance | actionTest1 | sampleAll | setNexthopGroup | setVrf | deny | setIdentityTag | actionSet | permit | actionTest3 | actionGoto | mirror | setAggregationGroup | useVrfSecondary | stripHeaderBytes | count | accessList | setTimestampHeader | reportFlow | maxActionEnum | setVrfSecondary | police | log | setNexthop | actionShaping | fwdAction*	class action type.
dscpValue	*integer*	DSCP value.

2.1560.1.1.8 SetTtlModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
ttlValue	*integer*	TTL value.

2.1560.1.1.9 TapAggActionSetModel

actionSet	*list of string*	List of action sets.
actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.10 TapAggDropActionModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.11 TapAggGroupModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
[aggGroup]	*list of string*	List of TapAgg Groups to flow the traffic.
[aggIntf]	*list of Interface name*	List of Tool Interfaces to flow the traffic.

2.1560.1.1.12 TapAggHeaderRemoveActionModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
preserveEth	*boolean*	Preserve the original Ethernet header, including VLAN tags if any.
size	*integer*	The number of bytes to remove after the Ethernet header, including any VLAN tags.

2.1560.1.1.13 TapAggIdTagModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
idTag	*integer*	Id-tag value.
innerIdTag	*integer*	Inner id-tag value.

2.1560.1.1.14 TapAggMacAddressActionModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
destMac	*MAC address*	Destination MAC address.
srcMac **	*MAC address*	Source MAC address. **Since: 4.27.2

2.1560.1.1.15 TapAggNexthopGroupModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
[nexthopGroup]	*list of string*	List of nexthop-groups to flow the traffic.

2.1560.1.1.16 TapAggStripHdrModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.
dot1qRemoveVlans	*string*	VLAN indices to be removed.
hdrType	*enum: noStrip* | *dot1q*	Header type.

2.1560.1.1.17 TapAggTimestampHeaderModel

actionType	*enum: setDscp* | *setTtl* | *sample* | *setHeaderRemove* | *sflow* | *setTc* | *setMacAddress* | *actionTest2* | *loadBalance* | *actionTest1* | *sampleAll* | *setNexthopGroup* | *setVrf* | *deny* | *setIdentityTag* | *actionSet* | *permit* | *actionTest3* | *actionGoto* | *mirror* | *setAggregationGroup* | *useVrfSecondary* | *stripHeaderBytes* | *count* | *accessList* | *setTimestampHeader* | *reportFlow* | *maxActionEnum* | *setVrfSecondary* | *police* | *log* | *setNexthop* | *actionShaping* | *fwdAction*	class action type.

2.1560.1.1.18 ActionSetModel

[actionSets]	*table(string) = NamedActionSetModel* see 2.1560.1.1.18.1	A mapping from action set name to actions.

2.1560.1.1.18.1 NamedActionSetModel

[actions]	*table(string) = ActionModel* see 2.1560.1.1.1	A mapping from action type to action value.
name	*string*	Action set name.

2.1560.1.2 PbrPolicyClassMapModel

[actionSetConfig] **	*ActionSetModel* see 2.1560.1.1.18	Action set configuration. **Since: 4.29.1
classMap	*ClassMapModel* see 2.241.1	Class map configuration.
[configuredAction]	*table(string) = ActionModel* see 2.1560.1.1.1	List of class actions.
name	*string*	Class map name.
[vrfs]	*table(string) = PbrPolicyClassMapVrfModel* see 2.1560.1.2.1	mapping of vrf name to policies.

2.1560.1.2.1 PbrPolicyClassMapVrfModel

action	*enum: dropAction | nexthopGroupAction | nexthopAction | noAction*	Type of routing action programmed for the vrf.
[actionVrf]	*string*	Name for egress vrf.
[nexthopGroup]	*string*	Nexthop group programmed for the vrf.
[nexthops]	*list of IpGenericAddr* see 2.296.1.1.1.1	Nexthops programmed for the vrf.
[ttl]	*integer*	TTL value.

2.1561 show policy-map type tapagg [...]

revision 2

see 2.1560

2.1562 show port numbering

numberingScheme	*enum: notAvailable | unknown | qsfpExpanded | qsfpDense | default | OCTAL4 | qsfpSparse*	Port numbering scheme.

2.1563 show port-channel [...]

see 2.325

2.1564 show port-channel dense

numberOfAggregators	*integer*	Number of aggregators.
numberOfChannelsInUse	*integer*	Number of channels in use.
portChannels	*table(Interface name) = PortChannelSummary* see 2.1564.1	Port-channels summary.

2.1564.1 PortChannelSummary

[fallback]	*FallbackConfigStatus* see 2.1564.1.1	Lag fallback status.
inactive **	*boolean*	Exceeds config limit. **Since: 4.25.0
[lacpMode]	*enum: off | passive | active*	LACP active or passive mode.
linkState	*enum: down | up*	state - port channels is up or down.
ports	*table(Interface name) = PortChannelMember-Summary* see 2.1564.1.2	Interfaces in port channel.
protocol	*enum: lacp | unconfigured | static*	Lag mode configuration.

2.1564.1.1 FallbackConfigStatus

[config]	*enum: fallbackStatic | fallbackIndividual*	Lag fallback type.
[status]	*enum: fallbackStatic | fallbackIndividual*	Lag fallback type.

2.1564.1.2 PortChannelMemberSummary

[exceedsMaxWeight] **	*boolean*	Exceeds maximum allowed weight. **Since: 4.26.2
intf	*interface name*	Interface name.
[lacpMisconfig]	*LacpMisconfig* see 2.1564.1.2.1	LACP misconfiguration state.
[lacpState]	*LacpPortState* see 2.839.1.2	LACP port state.
lagMember	*boolean*	Active member of the LAG.
linkDown	*boolean*	Is the interface down?
staticLag	*boolean*	Port is configured in a static LAG.
suspended	*boolean*	Interface is not active in the Port-Channel (wrong speed).

2.1564.1.2.1 LacpMisconfig

status	*enum: idle | unknown | mismatchAgg | negotiation | standby | noAgg | bundled*	LACP misconfiguration flags.

2.1565 show port-channel limits

portChannelLimits	*table(string) = LagGroupLimits* see 2.1565.1	Map a LAG Group name to the ports it applies to, and the hardware lag limits applicable to them.

2.1565.1 LagGroupLimits

maxPortChannels	*integer*	Maximum number of active port-channels.
maxPortsPerPortChannel	*integer*	Maximum number of active ports allowed in a port-channel.
portList	*list of Interface name*	Compatible ports.

2.1566 show port-channel load-balance (jericho | jericho2) fields [...]

revision 3

see 2.867

2.1567 show port-channel load-balance [...]

portChannels	*table(Interface name) = PortChannelTraffic* see 2.1567.1	Map the port-channel to interface traffic distribution.

2.1567.1 PortChannelTraffic

interfaceTraffic	*table(Interface name) = InterfaceTraffic* see 2.1567.1.1	Map from physical interface name to packet categories.

2.1567.1.1 InterfaceTraffic

inBroadcastPkts	*floating point number*	Broadcast packets received on this physical port, expressed as a percentage of total port-channel broadcast packets.
inMulticastPkts	*floating point number*	Multicast packets received on this physical port, expressed as a percentage of total port-channel multicast packets.
inUcastPkts	*floating point number*	Unicast packets received on this physical port, expressed as a percentage of total port-channel unicast packets.
outBroadcastPkts	*floating point number*	Broadcast packets sent on this physical port, expressed as a percentage of total port-channel broadcast packets.
outMulticastPkts	*floating point number*	Multicast packets sent on this physical port, expressed as a percentage of total port-channel multicast packets.
outUcastPkts	*floating point number*	Unicast packets sent on this physical port, expressed as a percentage of total port-channel unicast packets.

2.1568 show port-channel load-balance distribution

`globalDlbLagEnabled`	*boolean*	Dynamic load balancing on LAG is enabled globally.

2.1569 show port-channel load-balance trident fields

hashFunction	*enum: CRC16CCITT | CRC32HI | CRC32LO*	Hashing function implemented on this platform.
hashSeed	*integer*	hash seed.
lagHashFieldsCommon	*LagHashFieldsCommon* see 2.1569.1	Lag hashing on MISC fields.
lagHashFieldsGre	*LagHashFieldsGre* see 2.1569.2	Lag hashing on GRE fields.
lagHashFieldsGtp	*LagHashFieldsGtp* see 2.1569.3	Lag hashing on GTP fields.
lagHashFieldsIp	*LagHashFieldsIp* see 2.1569.4	Lag hashing on IP fields.
lagHashFieldsIpv6	*LagHashFieldsIpv6* see 2.1569.5	Lag hashing on IPV6 fields.
lagHashFieldsMac	*LagHashFieldsMac* see 2.1569.6	Lag hashing on MAC fields.
lagHashFieldsMim	*LagHashFieldsMim* see 2.1569.7	Lag hashing on MIM fields.
lagHashFieldsMpls	*LagHashFieldsMpls* see 2.1569.8	Lag hashing on MPLS fields.

2.1569.1 LagHashFieldsCommon

sysMacAddr	*boolean*	Lag hashing on the system MAC address.

2.1569.2 LagHashFieldsGre

ip6Ip6Outer	*boolean*	Lag hashing on GRE IPv6 over IPv6 outer IP address.
ip6IpOuter	*boolean*	Lag hashing on GRE IPv6 over IPv4 outer IP address.
ipIp6Outer	*boolean*	Lag hashing on GRE IPv4 over IPv6 outer IP address.
ipIpOuter	*boolean*	Lag hashing on GRE IPv4 over IPv4 outer IP address.

2.1569.3 LagHashFieldsGtp

gtpVersion	*string*	Hash on either GTP version 1 or 2.
teid	*boolean*	Lag hashing on GTP teid.

2.1569.4 LagHashFieldsIp

Could also be an instance of:
LagHashFieldsIpv6 2.1569.5

dstIpHashable	*boolean*	Lag hashing on destination IP address.
dstIpPortHashable	*boolean*	Lag hashing on destination IP port.
ingressIntfHashable	*boolean*	Lag hashing on ingress interface.
ipProtocolHashable	*boolean*	Lag hashing on the IP Protocol.

l34HeaderHashable	*boolean*	Lag hashing on layer 3 an 4 headers.
macHeader	*boolean*	MAC header exists.
srcIpHashable	*boolean*	Lag hashing on source IP address.
srcIpPortHashable	*boolean*	Lag hashing on source IP port.
symmetricHash	*boolean*	Symmetric hashing in use.
ttlHashable	*boolean*	Lag hashing on TTL/next hop.

2.1569.5 LagHashFieldsIpv6

dstIpHashable	*boolean*	Lag hashing on destination IP address.
dstIpPortHashable	*boolean*	Lag hashing on destination IP port.
flowLabel	*boolean*	Lag hashing on flow label.
ingressIntfHashable	*boolean*	Lag hashing on ingress interface.
ipProtocolHashable	*boolean*	Lag hashing on the IP Protocol.
l34HeaderHashable	*boolean*	Lag hashing on layer 3 an 4 headers.
macHeader	*boolean*	MAC header exists.
srcIpHashable	*boolean*	Lag hashing on source IP address.
srcIpPortHashable	*boolean*	Lag hashing on source IP port.
symmetricHash	*boolean*	Symmetric hashing in use.
ttlHashable	*boolean*	Lag hashing on TTL/next hop.

2.1569.6 LagHashFieldsMac

ethTypeHashable	*boolean*	Lag hashing on ether type.
ingressIntfHashable	*boolean*	Lag hashing on ingress interface.
macDstHashable	*boolean*	Lag hashing on destination MAC address.
macSrcHashable	*boolean*	Lag hashing on source MAC address.

2.1569.7 LagHashFieldsMim

isidHashable	*boolean*	Lag hashing on MIM ISID.
macDstHashable	*boolean*	Lag hashing on MIM destination MAC address.
macSrcHashable	*boolean*	Lag hashing on MIM source MAC address.
outerMac	*boolean*	Lag hashing on outer MAC.

2.1569.8 LagHashFieldsMpls

ipDstHashable	*boolean*	Lag hashing on IP destination address.
ipSrcHashable	*boolean*	Lag hashing on IP source address.
labelHashable	*boolean*	Lag hashing on MPLS label.
outerMac	*boolean*	Lag hashing on outer MAC.

2.1570 show port-channel load-balance trident udf

udfConfig	*list of StrataLagUdfHashLoadBalanceModel* see 2.1570.1	list of user defined field hashes.

2.1570.1 StrataLagUdfHashLoadBalanceModel

etherType	*enum: IPv6 | IPv4*	Ethernet type.
headerStr	*string*	known header to start the offset.
ipProto	*string*	IP protocol.
mask	*integer*	mask for the offset.
offset	*integer*	packet offset in bytes from the known header.
status	*boolean*	HW status of the configuration.

2.1571 show port-channel summary

see 2.1564

2.1572 show port-channel traffic [...]

see 2.1567

2.1573 show port-security

persistence **	*boolean*	Port security is persistent. **Since: 4.25.0
portStatistics	*table(Interface name) = PortStatistic* see 2.1573.1	Port Security Statistics.
secureAddressAging **	*boolean*	A secured address can age out. **Since: 4.25.0
secureAddressMoves **	*boolean*	A secured address can move. **Since: 4.25.0
totalAddresses	*integer*	Total number of addresses in the system.

2.1573.1 PortStatistic

currentAddr	*integer*	Current number of address learned.
maxSecureAddr	*integer*	Maximum number of learned addresses.
numberOfViolations	*integer*	Number of security violations that have occurred.
securityAction	*enum: shutdown | none | protect*	Action to take to protect the ports security.

2.1574 show port-security address

addresses	*list of Address* see 2.1574.1	Statistics for MAC addresses under port-security for this criteria.
totalAddresses	*integer*	Total number of MAC address under port-security for this criteria.

2.1574.1 Address

entryType	*enum: secureDynamic | secureConfigured*	Type of secured host.
interface	*interface name*	Interface that the address is learned on.
macAddress	*MAC address*	MAC address of host.
[remainingAge]	*integer*	Remaining minutes before interface ages out.
vlan	*integer*	VLAN host is learned in.

2.1575 show port-security interface [...]

interfaces	*table(Interface name) = Interface* see 2.1575.1	Per interface breakdown of port-security information.

2.1575.1 Interface

[addressChanges]	*integer*	Number of learn/move/age events on the interface.
agingTime	*integer*	Time before addresses age out (in minutes).
agingType	*inactivity*	Criteria for an address aging.
[allowedAddresses] **	*list of string*	Which Address:VlanId are allowed in protect mode. **Since: 4.15.3
[configuredMacAddresses]	*integer*	Number of configured MAC addresses on the interface.
[lastChangeDetails]	*InterestingAddress* see 2.1575.1.1	Last changed source address information.
[lastViolation]	*InterestingAddress* see 2.1575.1.1	Last violating address information.
logAddrsAfterLimit **	*enum: disabled | enabled*	Log new addresses seen past limit in protect mode. **Since: 4.15.3
maxMacAddresses	*integer*	Maximum MAC addresses allowed on the interface.
persistence **	*boolean*	Port security is persistent. **Since: 4.25.0
[portMaxEnabled] **	*boolean*	Port level limit is enabled. **Since: 4.25.1
portSecurityEnabled	*boolean*	Is port security enabled.
portStatus	*enum: secure-protected | secure-down | secure-up | secure-shutdown*	The status of the interface under port security.
secureAddressAging **	*boolean*	A secured address can age out. **Since: 4.25.0
secureAddressMoves **	*boolean*	A secured address can move. **Since: 4.25.0
secureStaticAddressAging	*disabled*	Are Secure Static addresses allowed to age (currently unused).
[securityViolationCount]	*integer*	Number of security violations on the interface.
[totalMacAddresses]	*integer*	Number of addresses learned on the interface.
violationMode	*enum: shutdown | none | protect*	Action port security will take to stop a violation.

2.1575.1.1 InterestingAddress

macAddress	*MAC address*	MAC Address.
time	*floating point number*	UTC representation of when the event occurred.

vlan	*integer*	VLAN the MAC address is in.

2.1576 show port-security mac-address

see 2.1574

2.1577 show port-security vlan [...]

interfaces	*table(Interface name) = VlanDict* see 2.1577.1	Per interface breakdown of vlan based port-security information.
totalAddresses	*integer*	Total number of addresses in the system.

2.1577.1 VlanDict

vlans	*table(numeric string) = VlanStatistic* see 2.1577.1.1	Per vlan breakdown of vlan based port-security information.

2.1577.1.1 VlanStatistic

action	*enum: shutdown | none | protect*	Action to take to protect the ports security.
maxAddrs	*integer*	Maximum number of learned addresses.
numAddrs	*integer*	Current number of learned addresses.
numViolations	*integer*	Number of security violations that have occurred on the VLAN.

2.1578 show priority-flow-control [...]

revision 3

see 2.458

2.1579 show priority-flow-control counters [...]

revision 3

see 2.459

2.1580 show priority-flow-control counters detail [...]

revision 3

see 2.460

2.1581 show priority-flow-control counters history [...]

interfaces	*table(Interface name) = PfcIntfHistoryCounters-Model* see 2.1581.1	A mapping of interface name to priority flow control history counters.
lastClearTimestamp	*floating point number*	UTC timestamp when clear command was last issued.

2.1581.1 PfcIntfHistoryCountersModel

priorities	*table(numeric string) = PfcHistoryCountersModel* see 2.1581.1.1	A mapping of priority value to priority flow control history counters.

2.1581.1.1 PfcHistoryCountersModel

[maxRxPfcOnTimeNs]	*integer*	Max RX PFC on duration in ns.
[maxRxPfcOnTimestamp]	*floating point number*	UTC timestamp when max RX PFC on duration was recorded.
[totalRxPfcOnTimeNs]	*integer*	Total RX PFC on duration in ns.
[totalRxPfcOnTransitions]	*integer*	Total RX PFC on transitions.

2.1582 show priority-flow-control counters watchdog [...]

revision 2

see 2.461

2.1583 show priority-flow-control counters watchdog drop [...]

revision 3

see 2.462

2.1584 show priority-flow-control interfaces Fabric status

see 2.466

2.1585 show priority-flow-control status [...]

revision 2

see 2.463

2.1586 show privilege

privilegeLevel	*integer*	The current privilege level.

2.1587 show processes

processes	*table(numeric string) = ProcessInfo* see 2.1587.1	Mappping of a process pid to the detailed process information.
timeInfo	*SystemTimeInfo* see 2.1587.2	uptime and load averages info.

2.1587.1 ProcessInfo

cmd	*string*	the name of the command that launched the process.
cpuPct	*floating point number*	the percentage of CPU used by the process.
memPct	*floating point number*	the percentage of memory used by the process.
startTime	*floating point number*	the starting time of the process.
state	*string*	the process state code.
totalActiveTime	*integer*	the length of time the process has been active.
ttyName	*string*	the terminal associated with the process.

2.1587.2 SystemTimeInfo

currentTime	*floating point number*	the current timestamp.
loadAvg	*list of floating point number*	the average number of processes in the system that are in a runnable or uninterruptable state for the past 1, 5, and 15 minutes.
upTime	*floating point number*	the length of time the system has been running.
users	*integer*	the number of users that are currently logged on.

2.1588 show processes top once [...]

cpuInfo	*table(string) = CpuInfo* see 2.1588.1	Mapping of each CPU to its state percentages based on the interval since the last refresh.
memInfo	*SystemMemSum* see 2.1588.3	System physical and virtual memory usage.
processes	*table(numeric string) = ProcessActivityInfo* see 2.1588.2	Mapping of a process pid tothe detailed process activity.
threadsStateInfo	*ThreadsInfo* see 2.1588.4	the number of threads for each state.
timeInfo	*SystemTimeInfo* see 2.1587.2	uptime and load averages info.

2.1588.1 CpuInfo

hwIrq	*floating point number*	the percentage of time spent servicing hardware interrupts.
idle	*floating point number*	the percentage of idle time.
ioWait	*floating point number*	the percentage of time waiting for I/O completion.
nice	*floating point number*	the percentage of time running niced user processes.
stolen	*floating point number*	the percentage of time stolen from this vm by the hypervisor.
swIrq	*floating point number*	the percentage of time spent servicing software interrupts.
system	*floating point number*	the percentage of time running kernel processes.
user	*floating point number*	the percentage of time running un-niced user processes.

2.1588.2 ProcessActivityInfo

activeTime	*integer*	the length of time the process has been active in Centiseconds.
cmd	*string*	the command that launched the process.
cpuPct	*floating point number*	the percentage of CPU used by the process.
cpuPctType	*string*	CPU percentage display format.
memPct	*floating point number*	the percentage of memory used by the process.
niceValue	*integer*	the nice value of the process corresponding to a user-space concept.
priority	*string*	the process priority.
residentMem	*string*	the resident memory size.
sharedMem	*string*	the shared memory used by the process.
status	*string*	the process status.
userName	*string*	the effective user name of the owner of the processes.
virtMem	*string*	the amount of virtual memory used by the process.

2.1588.3 SystemMemSum

physicalMem	*MemInfo* see 2.1588.3.1	system physical memory info.
swapMem	*MemInfo* see 2.1588.3.1	system virtual memory info.

2.1588.3.1 MemInfo

memBuffer	*integer*	the amount of memory has been used in Kbytes.
memFree	*integer*	the amount of unused memory in Kbytes.
memTotal	*integer*	the amount of total memory in Kbytes.
memUsed	*integer*	the amount of memory currently being used in Kbytes.

2.1588.4 ThreadsInfo

run	*integer*	the total number of running tasks.
sleep	*integer*	the total number of sleeping tasks.
stop	*integer*	the total number of stopped tasks.
total	*integer*	the number of total tasks or threads.
zombie	*integer*	the total number of zombie tasks.

2.1589 show ptp

revision 2

see 2.970

2.1590 show ptp ((local-clock time-properties) | time-property)

[message]	*string*	Message to display in case of unexpected state.
[timeProperty]	*PtpTimeProperty* see 2.1590.1	Time property data.

2.1590.1 PtpTimeProperty

currentUtcOffset	*integer*	Current offset between PTP time and UTC time. See IEEE 1588-2008 for more information.
currentUtcOffsetValid	*boolean*	Current offset between PTP time and UTC time is valid. See IEEE 1588-2008 for more information.
frequencyTraceable	*boolean*	timePropertiesDS.frequencyTraceable. See IEEE 1588-2008 for more information.

`leap59`	*boolean*	Last minute of today has 59 seconds. See IEEE 1588-2008 for more information.
`leap61`	*boolean*	Last minute of today has 61 seconds. See IEEE 1588-2008 for more information.
`ptpTimescale`	*boolean*	GM is using PTP timescale. See IEEE 1588-2008 for more information.
`timeSource`	*integer*	Time source GM is using. See IEEE 1588-2008 for more information.
`timeTraceable`	*boolean*	timePropertiesDS.timeTraceable. See IEEE 1588-2008 for more information.

2.1591 show ptp (clock | local-clock)

[ptpMode]	*enum: ptpOrdinaryMasterClock | ptpOneStepEndToEnd-TransparentClock | ptpDisabled | ptpGeneralized | ptpPeerTo-PeerTransparentClock | ptpBoundaryClock | ptpOneStepBoundaryClock | ptpEndToEndTransparent-Clock*	PTP mode.
[ptpOrdinaryOrBoundary-Clock]	*PtpOrdinaryOrBoundary-Clock* see 2.1591.1	Information for ordinary or boundary clocks.

2.1591.1 PtpOrdinaryOrBoundaryClock

[boundaryClockProperties]	*PtpBoundaryClock-Properties* see 2.1591.1.1	Boundary clock properties.
clockIdentity	*string*	DefaultDS clock identity.
clockQuality	*PtpClockQuality* see 2.1591.1.2	DefaultDS clock quality.
domainNumber	*integer*	DefaultDS domain number.
[localPriority] **	*integer*	DefaultDS local priority. See ITU-T G8275 for more information. **Since: 4.22.0
numberPorts	*integer*	DefaultDS number ports.
priority1	*integer*	DefaultDS priority1.
priority2	*integer*	DefaultDS priority2.

2.1591.1.1 PtpBoundaryClockProperties

[currentPtpSystemTime]	*integer*	Current sytem time measured with PTP.
[gptpCumulativeRateOffset]	*integer*	Estimated rate ratio expressed as an integer.
[gptpRateRatio]	*floating point number*	Estimated ratio of gPTP grandmaster to local clock frequency.
[lastSyncTime]	*integer*	Last time local clock synced from master.
meanPathDelay	*integer*	CurrentDS mean path delay to master in nanoseconds.
[neighborRateRatio]	*floating point number*	Estimated gPTP neighbor rate ratio.
offsetFromMaster	*integer*	CurrentDS offset from master in nanoseconds.
[skew]	*floating point number*	Ratio of master's seconds to local seconds.
stepsRemoved	*integer*	CurrentDS steps removed from GM.

2.1591.1.2 PtpClockQuality

accuracy	*integer*	Clock accuracy. See IEEE 1588-2008 for more information.
clockClass	*integer*	Clock class. See IEEE 1588-2008 for more information.
offsetScaledLogVariance	*integer*	Clock stability. See IEEE 1588-2008 for more information.

2.1592 show ptp (masters | parent)

[message]	*string*	Message to display in case of unexpected state.
[parent]	*PtpParent* see 2.1592.1	PTP parent information.

2.1592.1 PtpParent

gmClockIdentity	*string*	Grandmaster clock identity. See IEEE 1588-2008 for more information.
gmClockQuality	*PtpClockQuality* see 2.1591.1.2	Grandmaster clock quality. See IEEE 1588-2008 for more information.
gmPriority1	*integer*	Grandmaster priority1. See IEEE 1588-2008 for more information.
gmPriority2	*integer*	Grandmaster priority2. See IEEE 1588-2008 for more information.
parentClockIdentity	*string*	Parent clock identity. See IEEE 1588-2008 for more information.
[parentIp6Addr] **	*IPv6 address*	IPv6 address of the parent. **Since: 4.21.3
[parentIpAddr]	*IPv4 address*	IPv4 address of the parent.
parentPortNumber	*integer*	Parent PTP port number. See IEEE 1588-2008 for more information.
parentTwoStepFlag **	*boolean*	Parent PTP Boundary Clock Two-Step Flag. **Since: 4.29.0

2.1593 show ptp foreign-master-record

revision 2

intfForeignMasterRecords	*table(Interface name)* = *PtpIntfVlanForeignMaster-Records* see 2.1593.1	Collection of interface foreign master records.
[message]	*string*	Message to display in case of unexpected state.

2.1593.1 PtpIntfVlanForeignMasterRecords

interface	*interface name*	PTP interface name.
intfForeignMasterRecords	*list of PtpIntfForeignMasterRecords* see 2.1593.1.1	List of foreign master records for this interface. They might be on different VLANs when ptp vlan is configured on a trunk interface.

2.1593.1.1 PtpIntfForeignMasterRecords

foreignMasterRecords	*list of PtpForeignMasterDS* see 2.1593.1.1.1	List of foreign master records for this interface.
maxForeignRecords	*integer*	Most foreign masters heard from on this port.
[vlanId]	*integer*	VLAN on which the PTP trunk interface have receive foreign master records.

2.1593.1.1.1 PtpForeignMasterDS

foreignMasterAnnounceMessages	*integer*	Number of announce messages received from the foreign master.
[foreignMasterIp6Addr]	*IPv6 address*	IPv6 address of the foreign master.
[foreignMasterIpAddr]	*IPv4 address*	IPv4 address of the foreign master.
foreignMasterPortIdentity	*PtpPortIdentity* see 2.1593.1.1.1.1	PTP port identity on which the foreign master was seen.
timeAgo	*floating point number*	Last time this record was updated in monotonic seconds.

2.1593.1.1.1.1 PtpPortIdentity

clockIdentity	*string*	DefaultDS clock identity.
portNumber	*integer*	PTP port number.

2.1594 show ptp hardware-sync

see 2.1335

2.1595 show ptp hold-ptp-time

[holdPtpTimeExpiry] **	*floating point number*	Timestamp of when Ptp is expected to exit hold-over mode. **Since: 4.25.0
[holdPtpTimeInterval]	*integer*	hold-ptp-time interval in seconds.
[holdPtpTimeSinceEpoch] **	*floating point number*	Timestamp of when Ptp entered hold-over mode. **Since: 4.25.0
[holdPtpTimeState] **	*enum: holdoverInit* | *holdoverTimerExpired* | *holdoverMasterDetected* | *holdoverMode*	current Ptp hold-over mode. **Since: 4.25.0

[ptpMode]	*enum: ptpOrdinaryMasterClock* | *ptpOneStepEndToEnd-TransparentClock* | *ptpDisabled* | *ptpGeneralized* | *ptpPeerTo-PeerTransparentClock* | *ptpBoundaryClock* | *ptpOneStepBoundaryClock* | *ptpEndToEndTransparent-Clock*	PTP mode.

2.1596 show ptp interface INTF vlan VLAN_ID

revision 3

interfaces	*table(Interface name) = PtpIntfVlans* see 2.1596.1	Collection of interfaces to display.

2.1596.1 PtpIntfVlans

interface	*interface name*	PTP interface name.
ptpInterfaceVlansStates	*list of PtpIntfWrapper* see 2.1596.1.1	List of PTP states on a same interface. There can be several of them when PTP is configured to run on multiple VLANs on a trunk interface.

2.1596.1.1 PtpIntfWrapper

[interface]	*PtpIntfBase* see 2.1596.1.1.1	PTP interface state.

2.1596.1.1.1 PtpIntfBase

Could also be an instance of:
PtpIntf 2.1596.1.1.2
PtpMpassIntf 2.1596.1.1.3

[mpassDisabledReason]	*enum: notBoundaryMode | mlagIntfDisabled | notMlagInterface | ptpDisabledPeerLink | notConfigured | ptpDisabledInterface | ucastNegEnabled | delayMechanismNotE2e*	MPASS Disabled Reason.
[mpassEnabled]	*boolean*	Interface is PTP MPASS enabled.
[mpassStatus]	*boolean*	Interface in MPASS active state.
[portState]	*enum: psMaster | psPassive | psPreMaster | psDisabled | psIsolatedSlave | psSlave | psFaulty | psListening | psInitializing | psUncalibrated*	PTP port state. See IEEE 1588-2008 for more information.
[syncTestRole]	*enum: syncTestSlave | syncTestDisabled | syncTestMaster*	PTP synctest state.
[vlanId]	*integer*	VLAN on which the PTP trunk interface is configured.

2.1596.1.1.2 PtpIntf

adminDisabled	*boolean*	Interface is PTP disabled.
[announceInterval]	*floating point number*	Announce interval in seconds. See IEEE 1588-2008 for more information.

[announceReceiptTimeout]	*integer*	Announce receipt timeout, measured in announce intervals. See IEEE 1588-2008 for more information.
[asCapable]	*boolean*	Interface is gptp capable.
[counters]	*PtpIntfCountersWrapper* see 2.1596.1.1.2.1	Interface PTP counters.
[delayMechanism]	*string*	PTP delay mechanism. See IEEE 1588-2008 for more information.
[delayReqInterval]	*floating point number*	Delay request interval. See IEEE 1588-2008 for more information.
[dscpEvent]	*integer*	Dscp value to set for Ptp Ipv4 event packets.
[dscpGeneral]	*integer*	Dscp value to set for Ptp Ipv4 general packets.
[lastGptpResidenceTime]	*integer*	Minimum measured gptp residence time.
[localPriority]	*integer*	Local Priority value. See ITU-T G8275 for more information.
[maxGptpResidenceTime]	*integer*	Last measured gptp residence time.
[mcastTxAddr]	*MAC address*	Multicast destination address under PTP profile g8275.1.
[minGptpResidenceTime]	*integer*	Maximum measured gptp residence time.
[mpassDisabledReason]	*enum: notBoundaryMode* | *mlagIntfDisabled* | *notMlagInterface* | *ptpDisabledPeerLink* | *notConfigured* | *ptpDisabledInterface* | *ucastNegEnabled* | *delayMechanismNotE2e*	MPASS Disabled Reason.
[mpassEnabled]	*boolean*	Interface is PTP MPASS enabled.
[mpassStatus]	*boolean*	Interface in MPASS active state.
[peerDelayReqInterval]	*floating point number*	Peer delay request interval in seconds. See IEEE 1588-2008 for more information.
[peerPathDelay]	*integer*	Peer mean path delay in nanoseconds. See IEEE 1588-2008 for more information.
[portState]	*enum: psMaster* | *psPassive* | *psPreMaster* | *psDisabled* | *psIsolatedSlave* | *psSlave* | *psFaulty* | *psListening* | *psInitializing* | *psUncalibrated*	PTP port state. See IEEE 1588-2008 for more information.
ptpMode	*enum: ptpOrdinaryMasterClock* | *ptpOneStepEndToEnd-TransparentClock* | *ptpDisabled* | *ptpGeneralized* | *ptpPeerTo-PeerTransparentClock* | *ptpBoundaryClock* | *ptpOneStepBoundaryClock* | *ptpEndToEndTransparent-Clock*	PTP mode.

[syncInterval]	*floating point number*	Sync interval in seconds. See IEEE 1588-2008 for more information.
[syncReceiptTimeout]	*integer*	Sync receipt timeout, measured in sync intervals. See IEEE 802.1AS for more information.
[syncTestRole]	*enum: syncTestSlave | syncTestDisabled | syncTestMaster*	PTP synctest state.
[timeSinceAsCapableLastChanged]	*string*	Time elasped since AS capable state of the port last changed.
[transportMode]	*enum: ipv4 | ipv6 | layer2*	Type of packet used to send PTP messages.
[txLagMember]	*interface name*	Member interface used for sending packets in a LAG.
[vlanId]	*integer*	VLAN on which the PTP trunk interface is configured.

2.1596.1.1.2.1 PtpIntfCountersWrapper

[interfaceCounters]	*PtpIntfCountersBase* see 2.1596.1.1.2.2	PTP counters state.
[vlanId]	*integer*	VLAN on which the PTP trunk interface is configured.

2.1596.1.1.2.2 PtpIntfCountersBase

Could also be an instance of:
PtpIntfCounters 2.1596.1.1.2.3
PtpIntfMpassCounters 2.1596.1.1.2.4

2.1596.1.1.2.3 PtpIntfCounters

[announceRx]	*integer*	Announce messages rx.
[announceTx]	*integer*	Announce messages tx.
delayReqRx	*integer*	Delay request rx.
delayReqTx	*integer*	Delay request tx.
delayRespRx	*integer*	Delay response rx.
delayRespTx	*integer*	Delay response tx.
followUpRx	*integer*	Follow up rx.
followUpTx	*integer*	Follow up tx.
[gptpFollowUpMessagesTxMissed]	*integer*	Follow up gPTP messages tx missed.
[managementRx]	*integer*	Management rx.
[managementTx]	*integer*	Management tx.
pDelayReqRx	*integer*	Peer delay request rx.
pDelayReqTx	*integer*	Peer delay request tx.
pDelayRespFollowUpRx	*integer*	Peer delay response follow up rx.
pDelayRespFollowUpTx	*integer*	Peer delay response follow up tx.
pDelayRespRx	*integer*	Peer delay response rx.
pDelayRespTx	*integer*	Peer delay response tx.
[signalingRx]	*integer*	Signaling rx.

[signalingTx]	*integer*	Signaling tx.
syncRx	*integer*	Sync messages rx.
syncTx	*integer*	Sync messages tx.

2.1596.1.1.2.4 PtpIntfMpassCounters

announceLikeRx	*integer*	Forwarded Announce rx.
announceLikeTx	*integer*	Forwarded Announce tx.
delayReqLikeRx	*integer*	Forwarded Delay request rx.
delayReqLikeTx	*integer*	Forwarded Delay request tx.
delayRespLikeRx	*integer*	Forwarded Delay response rx.
delayRespLikeTx	*integer*	Forwarded Delay response tx.
followUpLikeRx	*integer*	Forwarded Follow up rx.
followUpLikeTx	*integer*	Forwarded Follow up tx.
syncLikeRx	*integer*	Forwarded Sync rx.
syncLikeTx	*integer*	Forwarded Sync tx.

2.1596.1.1.3 PtpMpassIntf

[counters]	*PtpIntfCountersWrapper* see 2.1596.1.1.2.1	Interface PTP MPASS counters.
delayReqExpiry	*floating point number*	Forwarded Delay request expiration timer.
[mpassDisabledReason]	*enum: notBoundaryMode* | *mlagIntfDisabled* | *notMlagInterface* | *ptpDisabledPeerLink* | *notConfigured* | *ptpDisabledInterface* | *ucastNegEnabled* | *delayMechanismNotE2e*	MPASS Disabled Reason.
[mpassEnabled]	*boolean*	Interface is PTP MPASS enabled.
[mpassStatus]	*boolean*	Interface in MPASS active state.
[portState]	*enum: psMaster* | *psPassive* | *psPreMaster* | *psDisabled* | *psIsolatedSlave* | *psSlave* | *psFaulty* | *psListening* | *psInitializing* | *psUncalibrated*	PTP port state. See IEEE 1588-2008 for more information.
syncExpiry	*floating point number*	Forwarded Sync expiration timer.
[syncTestRole]	*enum: syncTestSlave* | *syncTestDisabled* | *syncTestMaster*	PTP synctest state.
[vlanId]	*integer*	VLAN on which the PTP trunk interface is configured.

2.1597 show ptp interface [...]

revision 3

see 2.1596

2.1598 show ptp interface counters [...]

revision 3

interfaceCounters	*table(Interface name) = PtpIntfVlansCounters* see 2.1598.1	Interfaces to display counters.

2.1598.1 PtpIntfVlansCounters

interface	*interface name*	PTP interface name.
ptpInterfaceVlansCounters	*list of PtpIntfCountersWrapper* see 2.1596.1.1.2.1	List of PTP counters on a same interface. There can be several of them when PTP is configured to run on multiple VLANs on a trunk interface.

2.1599 show ptp interface counters drop [...]

counters	*table(Interface name) = PtpDebugIntfCounters* see 2.1599.1	PTP drop counters for all of the PTP interfaces.
disabled	*boolean*	PTP is disabled.

2.1599.1 PtpDebugIntfCounters

intfCounters	*list of PtpDebugCounter* see 2.1599.1.1	List of PTP drop counters on the interface.

2.1599.1.1 PtpDebugCounter

count	*integer*	Number of the PTP packets dropped.
direction	*enum: tx | rx*	Flow direction of the PTP packets dropped (Rx/Tx).
dropReason	*PtpDebugCounterDropReason* see 2.1599.1.1.1	Reason why the PTP packets are being dropped.
endpoint	*PtpEndpoint* see 2.1599.1.1.2	PTP endpoint where the packets are being dropped.
lastSeen **	*floating point number*	Last time a packet drop is seen, in seconds. **Since: 4.25.0

2.1599.1.1.1 PtpDebugCounterDropReason

debugReason	*enum: received* | *lessBoundaryHops* | *negativeTimestamp* | *staledPdelayReq* | *mpassSyncReceivedByPeer* | *mpassDelayReqReceived-ByPeer* | *oversizedMessage* | *mpassDisabled* | *missingUdpHdr* | *missingPtpHdr* | *vlanNotConfOnIntf* | *recvMyOwnClockMsg* | *badDestinedPort* | *blacklistedMasterIp* | *mpassNoLocalIntfReceiver* | *nonEthPktInGptp* | *serviceAcl* | *pamFailsTx* | *invalidGrantee* | *notFoundInitialPkt* | *arpNonExist* | *invalidIpv6* | *missingIpHdr* | *roleMaster* | *skewDropThreshold* | *badReqPort* | *noIngressSyncTimestamp* | *undersizedPkt* | *duplicateFollowUp* | *msgTxRxOnSameClk* | *invalidPdelayMsgType* | *no-EventMsgEgressTimestamp* | *mpassInvalidMessageType* | *badMaster* | *badMsgTypeInBC* | *invalidMgmtId* | *portNotSlave* | *invalidIpv4* | *MPDDropThreshold* | *unsupportedMsgTypeFor-Ucast* | *managementLoop* | *alreadySent* | *notBoundary* | *ordinaryMaster* | *noPamForTxIntf* | *notInUCast* | *notAristaEthernetType* | *waitForFollowUp* | *stepRemovedLarger255* | *noReqSent* | *unhealthyPortState* | *invalidUcastTlv* | *uCastGrantExpired* | *invalidDelayMsgType* | *invalidDmac* | *invalidEthernetType* | *unexpectedGrant* | *mpassNoMlagIntfStatus* | *mpassAnnounceBeforeSync* | *invalidTlv* | *mpassNoPeerLinkPortDS* | *mpassNoPeerLinkIntf* | *offsetFromMasterDrop-Threshold* | *pktQueueOverflow* | *unexpectedFollowUp* |	Reason of the PTP packet drop.

debugType	*enum: messageSignaling* | *messageFollowUpLike* | *messagePDelayResp* | *messageDelayReq* | *messagePDelayReq* | *messageSync* | *messageAnnounce* | *invalidPktFmt* | *messageP-DelayRespFollowup* | *messageDelayRespLike* | *messageAnnounceLike* | *messageDelayReqLike* | *messageSyncLike* | *messageFollowUp* | *messageDelayResp* | *messageManagement*	Type of the PTP packet drop reason.

2.1599.1.1.2 PtpEndpoint

domain	*integer*	Domain number of the PTP endpoint.
intf	*interface name*	Interface of the PTP endpoint.
ip	*IP address*	IP address of the PTP endpoint (only for PTP unicast negotiation).
vlanId	*integer*	VLAN of the PTP endpoint.

2.1600 show ptp interface vlans [...]

interfaces	*table(Interface name) = PtpVlansExtendedWrapper* see 2.1600.1	A mapping of interfaces to ptp configured vlans on trunk port.
[message]	*string*	Message in case of unexpected state.

2.1600.1 PtpVlansExtendedWrapper

notAllowedVlans	*list of integer*	PTP configured vlans but not allowed on trunk port.
notCreatedVlans	*list of integer*	PTP configured vlans but not created.
vlans	*list of integer*	PTP configured vlans on trunk port.

2.1601 show ptp ip access-list

see 2.53

2.1602 show ptp management-plane

revision 2

see 2.970

2.1603 show ptp monitor

[meanPathDelayThreshold]	*integer*	Mean path delay threshold.
monitorEnabled	*boolean*	PTP monitor enabled.
[offsetFromMasterThreshold]	*integer*	Offset from master threshold.
[ptpMode]	*enum: ptpOrdinaryMasterClock* | *ptpOneStepEndToEnd-TransparentClock* | *ptpDisabled* | *ptpGeneralized* | *ptpPeerTo-PeerTransparentClock* | *ptpBoundaryClock* | *ptpOneStepBoundaryClock* | *ptpEndToEndTransparent-Clock*	PTP mode.
ptpMonitorData	*list of PtpMonitorDataEntry* see 2.1603.1	Collection of PTP data from all interfaces.
[skewThreshold]	*floating point number*	Skew threshold.

2.1603.1 PtpMonitorDataEntry

intf	*interface name*	Slave interface on which the PTP data has been recorded.
lastSyncSeqId **	*integer*	Last sequenceId of sync / followUp PTP message processed. **Since: 4.21.3
meanPathDelay	*integer*	Mean path delay value.
offsetFromMaster	*integer*	Offset from master value.
realLastSyncTime	*integer*	Epoch timestamp of when the PTP data has been detected.
skew	*floating point number*	Ratio of master's second to local second.

2.1604 show ptp mpass

[delayReqTimeout]	*integer*	Forwarded Delay request expiration time-out in seconds.
ptpIntfSummaries	*table(Interface name) = PtpIntfVlanSummary* see 2.970.2	A mapping of interface to a summary of its MPASS configuration.
[syncTimeout]	*integer*	Forwarded Sync expiration timeout in messages.

2.1605 show ptp mpass interface [...]

revision 3

see 2.1596

2.1606 show ptp mpass interface counters [...]

revision 3

see 2.1598

2.1607 show ptp source

`[sourceIp]`	*IPv4 address*	Source IPv4 address to use when sending PTP messages.
`[sourceIp6]` **	*IPv6 address*	Source IPv6 address to use when sending PTP messages. **Since: 4.21.3

2.1608 show ptp source ip

see 2.1607

2.1609 show ptp unicast-negotiation candidate-grantor

`candidateGrantors`	*list of UcastNegCandidate-GrantorEntry* see 2.1609.1	List of candidate grantors.

2.1609.1 UcastNegCandidateGrantorEntry

`grantorState`	*enum: ucastNegMaster | ucastNegCandidateMaster | ucastNegBlacklisted*	State of the candidate grantor.
`intf`	*interface name*	Interface with which the candidate grantor is negotiating.
`ipAddr`	*IP address*	IP address of the candidate grantor.
`profileName`	*string*	Profile with which the candidate grantor is associated.

2.1610 show ptp unicast-negotiation granted

`interfaces`	*table(Interface name) = UcastNegMessages* see 2.1610.1	A mapping from the interface name to message infos.

2.1610.1 UcastNegMessages

messages	*list of UcastNegMessage* see 2.1610.1.1	Unicast Negotiation request sent from master/slave.

2.1610.1.1 UcastNegMessage

duration	*integer*	Duration of time, in seconds, agreed upon between the grantor and grantee.
expirationTime	*floating point number*	Time, in seconds, at which request was sent from grantor/grantee to Slave/Master.
ipAddr	*IP address*	Ip address of the Master/Slave from which we received a message.
logInterval	*integer*	log Interval of time, in seconds, agreed upon between the the and grantee.
messageType	*enum: messageSignaling* | *messageFollowUpLike* | *messagePDelayResp* | *messageDelayReq* | *messagePDelayReq* | *messageAnnounce* | *messagePDelayResp-Followup* | *messageDelayRespLike* | *messageAnnounceLike* | *messageDelayReqLike* | *messageSyncLike* | *messageFollowUp* | *messageSync* | *none* | *messageDelayResp* | *messageManagement*	Type of Message sent from potential Master/Slave to Slave/Master.

2.1611 show ptp unicast-negotiation profile [...]

profiles	*table(string) = UcastNegProfile* see 2.1611.1	Unicast Negotiation Profiles list.

2.1611.1 UcastNegProfile

announceDuration	*integer*	Announce duration of time of profile. See IEEE 1588-2008 for more information.
announceInterval	*integer*	Announce interval of time of profile. See IEEE 1588-2008 for more information.
delayResponseDuration	*integer*	Delay Response interval of profileSee IEEE 1588-2008 for more information.
delayResponseInterval	*integer*	Delay Response interval of profileSee IEEE 1588-2008 for more information.
syncDuration	*integer*	Sync duration of time of profile. See IEEE 1588-2008 for more information.
syncInterval	*integer*	Sync interval of time of profile. See IEEE 1588-2008 for more information.

2.1612 show ptp unicast-negotiation remote-grantee

remoteGrantees	*list of UcastNegRemoteGranteeEntry* see 2.1612.1	List of remote grantees.

2.1612.1 UcastNegRemoteGranteeEntry

intf	*interface name*	Interface with which the remote grantee can negotiate.
ipAddrAndMask	*IP address and mask length*	Allowed IP address range for the remote grantee.
profileName	*string*	Profile with which the remote grantee is associated.

2.1613 show ptp unicast-negotiation requested

see 2.1610

2.1614 show qos ecn [...]

delayEcnSupported	*boolean*	Global Delay based ECN supported.
[globalEcnDelayParameters]	*EcnDelayParametersModel* see 2.1614.1	Global delay based ECN Parameters. Not applicable if Global delay based ECN is disabled.
[globalEcnParameters]	*EcnParametersModel* see 2.1614.2	Global length based ECN Parameters. Not applicable if Global length based ECN is disabled.
globalEcnSupported	*boolean*	Global Buffer ECN supported.
globalEcnWeightSupported **	*boolean*	Global ECN Supported. **Since: 4.21.3

2.1614.1 EcnDelayParametersModel

[configuredThreshold] **	*floating point number*	Configured Threshold value in microseconds. **Since: 4.17.0
[source] **	*enum: global | none | local*	Configuration source. **Since: 4.17.0
threshold	*floating point number*	Actual Threshold value in microseconds.

2.1614.2 EcnParametersModel

[maxDroprate] **	*integer*	Mark probability rate. **Since: 4.21.3
maxThreshold	*integer*	Maximum Threshold for ECN.
minThreshold	*integer*	Minimum Threshold for ECN.
[operationalMaxDroprate] **	*integer*	Operational Mark probability rate. **Since: 4.27.0
[segmentSizeInBytes]	*integer*	Size of segment unit in bytes.

unit	*enum: segments | microseconds | bytes | kbytes | milliseconds | mbytes*	Threshold unit for ECN.
[weight] **	*integer*	Weight for averaging queue size. **Since: 4.21.3

2.1615 show qos ecn counters

chipEcnCounter	*table(string) = integer*	A mapping between a chip name and ECN counter.

2.1616 show qos ecn delay [...]

[ecnDelayParameters]	*EcnDelayParametersModel* see 2.1614.1	General delay ECN parameters.

2.1617 show qos interfaces [...]

intfAllQosAll	*table(Interface name) = IntfQosAllModel* see 2.1617.1	All the Interfaces with all Qos details for it.

2.1617.1 IntfQosAllModel

intfQosModel	*InterfaceQosModel* see 2.1617.1.1	Qos on Interface.
intfServicePolicyQosModel	*IntfServicePolicyQosModel* see 2.1617.1.2	Service Policy on Interface.

2.1617.1.1 InterfaceQosModel

burstSizeConfigSupported **	*boolean*	Burst-size config is supported. **Since: 4.24.0
[configuredBurstSize] **	*BurstSizeModel* see 2.1617.1.1.1	Configured burst-size. **Since: 4.24.0
[configuredPolicerPktSize-Adj] **	*PolicerPktSizeAdjModel* see 2.1617.1.1.4	Configured policer packet size adjustment. **Since: 4.27.0
[configuredPortGuaranteed-Bw] **	*GuaranteedBwModel* see 2.1617.1.1.2	Configured port guaranteed bandwidth. **Since: 4.25.2
[configuredPortShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured port shape rate.
[configuredScheduler-Compensation] **	*SchedulerCompensation-Model* see 2.1617.1.1.5	Configured scheduler compensation. **Since: 4.26.2
cosToTcProfile **	*string*	COS-to-TC profile. **Since: 4.22.1
cpuTcToCosNamedMap **	*string*	CPU TC-to-COS map. **Since: 4.23.2
[defaultCos]	*integer*	Default COS value, if supported.
defaultCosSupported	*boolean*	Is default COS supported.
[defaultDscp]	*integer*	Default DSCP value, if supported.
defaultDscpSupported	*boolean*	Is default DSCP supported.
dscpPreserveIpMplsEncapMode **	*boolean*	DSCP Preserve for MPLS IP encap. **Since: 4.28.2
[dscpRewriteActive]	*boolean*	DSCP Rewrite Active, if hardware supports reporting.
dscpRewriteMap **	*string*	TC-to-DSCP map. **Since: 4.23.2
[dscpToCosRewriteActive]	*boolean*	DSCP To COS Rewrite active status, if hardware supports reporting.
dscpToTcMap **	*string*	DSCP-to-TC map. **Since: 4.27.0
expToTcProfile **	*string*	EXP-to-TC profile. **Since: 4.30.1
interface	*interface name*	Qos on Interface.
interfaceGuaranteedBw-Supported **	*boolean*	Guaranteed bandwidth config is supported. **Since: 4.25.2
intfIsLag	*boolean*	Is a LAG Interface.

intfQosDetailedInfo-Available	*boolean*	Detailed Qos info of the interface is available.
intfQosInfoAvailable	*boolean*	Interface Qos info availability.
[operationalBurstSize] **	*BurstSizeModel* see 2.1617.1.1.1	Operational burst-size. **Since: 4.24.0
[operationalPolicerPktSize-Adj] **	*PolicerPktSizeAdjModel* see 2.1617.1.1.4	Operational policer packet size adjustment. **Since: 4.27.0
[operationalPortGuaranteed-Bw] **	*GuaranteedBwModel* see 2.1617.1.1.2	Operational port guaranteed bandwidth. **Since: 4.25.2
[operationalPortShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Operational port shape rate.
[operationalScheduler-Compensation] **	*SchedulerCompensation-Model* see 2.1617.1.1.5	Operational scheduler compensation. **Since: 4.26.2
policerPktSizeAdjConfig-Supported **	*boolean*	Policer packet size config is supported. **Since: 4.27.0
schedulerCompensation-ConfigSupported **	*boolean*	Scheduler compensation config is supported. **Since: 4.26.2
[schedulingGroupModel] **	*IntfSchedulingGroupModel* see 2.1617.1.1.3	Scheduling Group Model. **Since: 4.25.0
[sliceName] **	*string*	Slice name corresponding to the Fabric interface. **Since: 4.17.0
tcToCosNamedMap **	*string*	TC-to-COS map. **Since: 4.23.2
trustMode	*enum: cosTrusted | dscpTrusted | unTrusted*	Qos Trustmode of interface.
[txQueueQosModel]	*TxQueueQosModel* see 2.1617.1.1.6	Transmit Queue Model.

2.1617.1.1.1 BurstSizeModel

unit	*enum: burstUnitMBytes | burstUnitKBytes | burstUnitBytes*	Burst-size unit.
value	*integer*	Burst-size value.

2.1617.1.1.2 GuaranteedBwModel

bandwidth	*integer*	Guaranteed bandwidth.
[percent] **	*integer*	Guaranteed bandwidth as percentage of reference. **Since: 4.25.0
[percentOf] **	*enum: unknown | subinterfaceShapeRate | lineRate | schedulingGroupShapeRate | parentInterfaceShapeRate*	Reference for guaranteed bandwidth percent. **Since: 4.25.2
unit	*enum: percent | kbps | unspecified | pps*	Guaranteed bandwidth unit.

2.1617.1.1.3 IntfSchedulingGroupModel

schedulingGroups	*table(string) = SchedulingElementModel* see 2.1617.1.1.3.1	Scheduling Groups configured for the interface, indexed by their name.

2.1617.1.1.3.1 SchedulingElementModel

Could also be an instance of:
HierarchyModel 2.1617.1.1.3.2
SchedulingGroupWithMembersModel 2.1617.1.1.3.3

[configuredGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Configured Guaranteed bandwidth.
[configuredShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured Shape rate.
[operationalGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Operational Guaranteed bandwidth.
[operationalShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Operational Shape rate.

2.1617.1.1.3.2 HierarchyModel

[configuredGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Configured Guaranteed bandwidth.
[configuredShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured Shape rate.
[interfaceSpeed]	*integer*	linkSpeed (bps).
name	*string*	Name of the entity in the hierarchy.
[operationalGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Operational Guaranteed bandwidth.
[operationalShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Operational Shape rate.

2.1617.1.1.3.3 SchedulingGroupWithMembersModel

[configuredGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Configured Guaranteed bandwidth.
[configuredShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured Shape rate.
[interfaces]	*table(Interface name) = SchedulingElementModel* see 2.1617.1.1.3.1	Member subinterfaces of the scheduling group.
[operationalGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Operational Guaranteed bandwidth.
[operationalShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Operational Shape rate.

2.1617.1.1.4 PolicerPktSizeAdjModel

profile	*string*	Packet size adjustment profile.

value	*integer*	Packet size adjustment configuration associated with the profile.

2.1617.1.1.5 SchedulerCompensationModel

unit	*burstUnitBytes*	Scheduler Compensation unit.
value	*integer*	Scheduler Compensation value.

2.1617.1.1.6 TxQueueQosModel

burstSizeConfigSupported **	*boolean*	Burst-size config is supported. **Since: 4.24.0
delayEcnSupported **	*boolean*	Delay based ECN supported. **Since: 4.17.0
ecnSupported	*boolean*	ECN supported.
guaranteedBwSupported	*boolean*	Guaranteed bandwidth supported.
multipleSchedGroups-Supported	*boolean*	Multiple scheduling groups supported.
nonEctSupported **	*boolean*	NON-ECT supported. **Since: 4.20.5
numTxQueue	*integer*	Number of Transmit Queues.
[numTxQueuePrint] **	*integer*	Number of Transmit Queues to print. **Since: 4.25.1
txQueueList	*list of TxQueue* see 2.1617.1.1.6.1	List of Transmit Queues.
wredSupported **	*boolean*	WRED supported. **Since: 4.17.0
wrrSupported	*boolean*	WRR Scheduling supported.

2.1617.1.1.6.1 TxQueue

[configuredBurstSize] **	*BurstSizeModel* see 2.1617.1.1.1	Configured burst-size. **Since: 4.24.0
[configuredGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Configured Guaranteed bandwidth.
[configuredSchedMode]	*SchedulingModeModel* see 2.1617.1.1.6.2	Configured Scheduling mode.
[configuredShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured Shape rate.
[configuredWrrBw]	*integer*	Configured WRR Bandwidth in percent.
ecnStatus	*enum: delayEnabled | disabled | not applicable | enabled | lengthAndDelayEnabled*	ECN Status on Transmit Queue.
nonEctStatus **	*enum: disabled | not applicable | enabled*	NON-ECT Status on Transmit Queue. **Since: 4.20.5
[operationalBurstSize] **	*BurstSizeModel* see 2.1617.1.1.1	Operational burst-size. **Since: 4.24.0
[operationalGuaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Operational Guaranteed bandwidth.

[operationalSchedMode]	*SchedulingModeModel* see 2.1617.1.1.6.2	Operational Scheduling mode.
[operationalShapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Operational Shape rate.
[operationalWrrBw]	*integer*	Operational WRR Bandwidth in percent.
schedGroupId	*integer*	Scheduling Group Id.
txQueue	*string*	Transmit Queue.
txQueuePriority	*integer*	Transmit Queue Priority - higher txQueuePriority value has precedence.
wredStatus **	*enum: disabled | not applicable | enabled*	WRED Status on Transmit Queue. **Since: 4.17.0

2.1617.1.1.6.2 SchedulingModeModel

schedulingMode	*enum: roundRobin | strictPriority | unspecified*	Scheduling mode.

2.1617.1.2 IntfServicePolicyQosModel

intfPolicyMaps	*table(string) = PolicyQosModel* see 2.1617.1.2.1	Policy maps list for an interface for different directions.

2.1617.1.2.1 PolicyQosModel

directionString	*enum: input | output*	Direction in which policy is applied.
name	*string*	Name of the policy map.
status	*enum: hwPrgmStatusSuccess | hwPrgmStatusFail | hwPrgmStatusInProgress*	Hardware programming status.

2.1618 show qos interfaces drop-thresholds [...]

dropThresholdsSupported	*boolean*	Drop Precedence thresholds supported.
intfDropThresholds-Collection	*table(Interface name) = IntfDropThresholdsModel* see 2.1618.1	A mapping of an interface to its drop thresholds.

2.1618.1 IntfDropThresholdsModel

txQueueDropThresholdsList	*list of TxQueueDrop-ThresholdsModel* see 2.1618.1.1	List of Transmit Queue Drop Threshold Settings.

2.1618.1.1 TxQueueDropThresholdsModel

dropThresholds	*table(numeric string) = integer*	Drop Thresholds per drop precedence in percent.
txQueue	*string*	Name of Egress Queue.

2.1619 show qos interfaces dscp mpls encapsulation disabled [...]

interfaces	*list of Interface name*	Interfaces with DSCP Preserve enabled.

2.1620 show qos interfaces ecn [...]

intfEcnCollection	*table(Interface name) = IntfEcnModel* see 2.1620.1	Interface ECN.

2.1620.1 IntfEcnModel

txQueueEcnList	*list of TxQueueEcnModel* see 2.1620.1.1	List of Transmit Queue ECN Settings.
txQueueNonEctList **	*list of TxQueueNonEctModel* see 2.1620.1.2	List of Transmit Queue NON-ECT Settings. **Since: 4.28.0

2.1620.1.1 TxQueueEcnModel

ecnMarkProbSupported **	*boolean*	ECN mark-probability supported. **Since: 4.21.3
ecnWeightSupported **	*boolean*	ECN Weight Supported. **Since: 4.21.3
txQueue	*string*	Name of Egress Queue.
[txQueueEcnDelayParameters] **	*EcnDelayParametersModel* see 2.1614.1	Transmit Queue Delay based ECN Parameters. Not Applicable if Transmit Queue delay ECN is disabled. **Since: 4.17.0
txQueueEcnDelaySupported **	*boolean*	Trasmit Queue delay ECN supported. **Since: 4.17.0
[txQueueEcnParameters]	*EcnParametersModel* see 2.1614.2	Transmit Queue ECN Parameters. Not Applicable if Transmit Queue buffer ECN is disabled.

2.1620.1.2 TxQueueNonEctModel

nonEctWeightSupported	*boolean*	NON-ECT Weight Supported.
txQueue	*string*	Name of Egress Queue.
[txQueueNonEctParameters]	*EcnParametersModel* see 2.1614.2	Transmit Queue NON-ECT Parameters. Not Applicable if Transmit Queue NON-ECT is disabled.

2.1621 show qos interfaces ecn counters queue [...]

intfQueueCounters	*table(Interface name) = EcnQueueCounterModel* see 2.1621.1	Mapping from interface to tx-queue ecn counters.

2.1621.1 EcnQueueCounterModel

queueCounters	*table(numeric string) = string*	Mapping from tx-queue to ecn counters.

2.1622 show qos interfaces fabric [...]

fabricAllQosAll	*table(string) = IntfQosAllModel* see 2.1617.1	A mapping between Fabric Interfaces to Qos details.

2.1623 show qos interfaces latency maximum [...]

interfaces	*table(Interface name) = IntfLatencyThresholdModel* see 2.1623.1	A mapping of interface to transmit queue latency threshold.

2.1623.1 IntfLatencyThresholdModel

txQueues	*list of TxQueueLatency-ThresholdModel* see 2.1623.1.1	List of transmit queue latency threshold settings.

2.1623.1.1 TxQueueLatencyThresholdModel

threshold	*integer*	Transmit queue latency threshold.
txQueue	*string*	Name of transmit queue.
unit	*string*	Latency threshold unit.

2.1624 show qos interfaces random-detect drop [...]

intfWredCollection	*table(Interface name) = IntfWredModel* see 2.1624.1	Interface WRED.

2.1624.1 IntfWredModel

txQueueWredList	*list of TxQueueWredModel* see 2.1624.1.1	List of Transmit Queue WRED Settings.

2.1624.1.1 TxQueueWredModel

dpWredSupported **	*boolean*	WRED per Drop-Precedence Supported. **Since: 4.27.2
txQueue	*string*	Name of Egress Queue.
[txQueueWredParameters]	*WredParametersModel* see 2.1624.1.1.1	Transmit Queue WRED Parameters. Not Applicable if Transmit Queue WRED is disabled.
wredWeightSupported **	*boolean*	WRED weight Supported. **Since: 4.21.3

2.1624.1.1.1 WredParametersModel

[dp] **	*integer*	Drop Precedence value. **Since: 4.27.2
maxDroprate	*integer*	Drop probability rate.
maxThreshold	*integer*	Maximum Threshold for WRED.
minThreshold	*integer*	Minimum Threshold for WRED.
[operationalMaxDroprate] **	*integer*	Operational Mark probability rate. **Since: 4.27.0
[segmentSizeInBytes]	*integer*	Size of segment unit in bytes.
unit	*enum: segments | microseconds | bytes | kbytes | milliseconds | mbytes*	Threshold unit for WRED.
[weight] **	*integer*	Weight for averaging queue size. **Since: 4.21.3

2.1625 show qos interfaces random-detect ecn [...]

Could also be an instance of:
IntfAllEcnCollectionModel 2.1620

intfEcnCollection	*table(Interface name) = IntfEcnModel* see 2.1620.1	Interface ECN.

see 2.1620

2.1626 show qos interfaces trust [...]

intfTrustModeCollection	*table(Interface name) = IntfTrustModeModel* see 2.1626.1	Interface trust mode.

2.1626.1 IntfTrustModeModel

[intfTrustModeParameters]	*IntfTrustModeParameters-Model* see 2.1626.1.1	Trust Mode Parameters.

2.1626.1.1 IntfTrustModeParametersModel

configuredTrustMode	*enum: DSCP | UNTRUSTED | COS | DEFAULT*	Configured trust mode.
operationalTrustMode	*enum: DSCP | UNTRUSTED | COS*	Operational trust mode.

2.1627 show qos interfaces wred counters queue [...]

intfQueueCounters	*table(Interface name) = WredQueueCounterModel* see 2.1627.1	Mapping from interface to tx-q wred counters.

2.1627.1 WredQueueCounterModel

queueCounters	*table(numeric string) = MetaWredQueueCounterModel* see 2.1627.1.1	Mapping from tx-queue to wred counters.

2.1627.1.1 MetaWredQueueCounterModel

[ecnBytes]	*integer*	ECN marked bytes.
[ecnPkts]	*integer*	ECN marked packets.
[enqueueBytes]	*integer*	Total enqueued bytes.
[enqueuePkts]	*integer*	Total enqueued packets.
[tailDropBytes]	*integer*	Tail dropped bytes.
[tailDropPkts]	*integer*	Tail dropped packets.
[wredDropBytes]	*integer*	Bytes dropped due to WRED.
wredDropPkts	*integer*	Packets dropped due to WRED.

2.1628 show qos map cos to traffic-class [...]

cosToTcProfiles	*table(string) = CosToTcProfileModel* see 2.1628.1	CoS to Traffic-Class profiles, indexed by their name.

2.1628.1 CosToTcProfileModel

cosToTcMap	*table(numeric string) = integer*	CoS to Traffic-Class mapping.

2.1629 show qos map dscp to traffic-class [...]

dscpToTcNamedMaps	*table(string) = DscpToTcMapModel* see 2.1629.1	DSCP To Traffic-Class maps, indexed by their name.

2.1629.1 DscpToTcMapModel

dscpToTcNamedMap	*table(numeric string) = integer*	DSCP To Traffic-Class mapping.

2.1630 show qos map exp to traffic-class [...]

expToTcProfiles	*table(string) = ExpToTcProfileModel* see 2.1630.1	EXP to Traffic-Class profiles, indexed by their name.

2.1630.1 ExpToTcProfileModel

expToTcMap	*table(numeric string) = integer*	EXP to Traffic-Class mapping.

2.1631 show qos map traffic-class to cos [...]

tcToCosNamedMaps	*table(string) = TcToCosMapModel* see 2.1631.1	Traffic-Class to CoS profiles, indexed by their name.

2.1631.1 TcToCosMapModel

tcToCosNamedMap	*table(string) = integer*	Traffic-Class to CoS mapping.

2.1632 show qos map traffic-class to dscp [...]

dscpRewriteMaps	*table(string) = DscpRewriteMapModel* see 2.1632.1	A mapping between Dscp RewriteMap names and Dscp RewriteMap.

2.1632.1 DscpRewriteMapModel

tcDpToDscpMap **	*table(string) = integer*	Traffic-Class, Drop-Precedence to DSCP mapping. **Since: 4.27.1
tcToDscpMap	*table(numeric string) = integer*	Traffic-Class to DSCP mapping.

2.1633 show qos maps [...]

cosRewriteDisableSupported	*boolean*	COS rewrite disable supported.
cosRewriteEnabled	*boolean*	COS rewrite enabled.
cosRewriteSupported	*boolean*	COS rewrite supported.
cosToTcMap	*table(numeric string) = integer*	COS to Traffic-Class mapping.
[cosToTcProfileAllModel] **	*CosToTcProfileAllModel* see 2.1628	User configured COS to Traffic-Class profiles. **Since: 4.22.1
cosToTcProfilePerInterface-Supported **	*boolean*	Configuration of CoS to Traffic-Class Map on interface supported. **Since: 4.22.1
deiToDpMap **	*table(numeric string) = integer*	DEI to Drop-Precedence mapping. **Since: 4.30.1
deiToDpMapSupported **	*boolean*	DEI to Drop-Precedence mapping supported. **Since: 4.30.1
dscpRewriteEnabled	*boolean*	DSCP rewrite enabled.
[dscpRewriteMapAllModel] **	*DscpRewriteMapAllModel* see 2.1632	User configured Traffic-Class and Drop-Precedence to DSCP profiles. **Since: 4.23.2
dscpRewritePerInterface-Supported **	*boolean*	Configuration of Traffic-Class to DSCP Map on interface supported. **Since: 4.23.2
dscpRewriteSupported	*boolean*	DSCP rewrite supported.
[dscpToDpMap] **	*table(numeric string) = integer*	DSCP to Drop-Precedence mapping. **Since: 4.22.0
dscpToDpMapSupported **	*boolean*	DSCP to Drop-Precedence mapping supported. **Since: 4.22.0
dscpToTcMap	*table(numeric string) = integer*	DSCP to Traffic-Class mapping.
[dscpToTcMapAllModel] **	*DscpToTcMapAllModel* see 2.1629	User configured Dscp to Traffic-Class maps. **Since: 4.25.1
dscpToTcMapPerSubInterface-Supported **	*boolean*	Configuration of DSCP To Traffic-Class Map on subInterface supported. **Since: 4.28.0
dynamicExpMappingSupported **	*boolean*	Configuration of EXP to Traffic-Class and Traffic-Class to EXP mapping supported. **Since: 4.22.0
[expToTcMap] **	*table(numeric string) = integer*	EXP to Traffic-Class mapping when dynamic EXP mapping is supported. **Since: 4.22.0
[expToTcProfileAllModel] **	*ExpToTcProfileAllModel* see 2.1630	User configured EXP to Traffic-Class profiles. **Since: 4.30.1
expToTcProfilePerInterface-Supported **	*boolean*	Configuration of EXP to Traffic-Class Map on interface supported. **Since: 4.30.1
numMulticastQueueSupported	*integer*	Number of Multicast Queues supported.
numTcSupported	*integer*	Number of Traffic Classes supported.

numTxQueueSupported	*integer*	Number of Transmit Queues supported.
pwDecapDscpQosMapSupported **	*boolean*	Configuration of DSCP To Traffic-Class Map on pseudowire supported. **Since: 4.25.1
[tcToCosMap]	*table(numeric string) = integer*	Traffic-Class to COS mapping when COS rewrite is supported.
[tcToCosMapAllModel] **	*TcToCosMapAllModel* see 2.1631	User configured Traffic-Class to CoS profiles. **Since: 4.23.2
tcToCosMapPerInterfaceSupported **	*boolean*	Configuration of Traffic-Class to CoS Map on interface supported. **Since: 4.23.2
[tcToDscpMap]	*table(numeric string) = integer*	Traffic-Class to DSCP mapping when DSCP rewrite is supported.
tcToDscpMapSupported	*boolean*	Traffic-Class to DSCP mapping supported.
[tcToExpMap] **	*table(numeric string) = integer*	Traffic-Class to EXP mapping when dynamic EXP mapping is supported. **Since: 4.22.0
[tcToMcTxQueueMap]	*table(numeric string) = integer*	Traffic-Class to mc-tx-queue mapping when multicast queues are supported.
[tcToPriorityGroupMap] **	*table(numeric string) = integer*	Traffic-Class to PriorityGroup mapping. **Since: 4.21.0
tcToPriorityGroupMapSupported **	*boolean*	Traffic-Class to Priority Group supported. **Since: 4.21.0
[tcToTxQueueMap]	*table(numeric string) = integer*	Traffic-Class to tx-queue mapping when unicast queues are supported.

2.1634 show qos profile [...]

qosProfiles	*table(string) = QosProfileModel* see 2.1634.1	A mapping between QoS profile name and QoS profiles.

2.1634.1 QosProfileModel

[configuredPortGuaranteed-Bw] **	*GuaranteedBwModel* see 2.1617.1.1.2	Configured port Guaranteed Bandwidth. **Since: **currentRelease**
[configuredPortShapeRate] **	*ShapeRateModel* see 2.477.1.1.1.2	Configured port shape rate. **Since: 4.17.0
[configuredScheduler-Compensation] **	*SchedulerCompensation-Model* see 2.1617.1.1.5	Configured port scheduler compensation. **Since: 4.26.2
[configuredServicePolicy] **	*PolicyQosModel* see 2.1617.1.2.1	Configured Service Policy. **Since: 4.18.0
[defaultCos] **	*integer*	Default COS value. **Since: 4.17.0
defaultCosSupported **	*boolean*	Default COS supported. **Since: 4.17.0
[defaultDscp] **	*integer*	Default DSCP value. **Since: 4.17.0
defaultDscpSupported **	*boolean*	Default DSCP supported. **Since: 4.17.0
dscpPreserveIpMplsEncapMode **	*boolean*	DSCP Preserve for MPLS IP encap. **Since: 4.28.2
numTxQueueSupported	*integer*	Number of Transmit Queues supported.
pfcEnabled	*boolean*	PFC enabled.
pfcPriorities	*table(numeric string) = PfcProfileModel* see 2.1634.1.1	A mapping between PFC priority and PFC configuration.
pfcWatchdogEnabled **	*boolean*	PFC Watchdog enabled. **Since: 4.17.0
[pfcWatchdogPortAction] **	*string*	Configured per-port action value for PFC Watchdog. **Since: 4.20.0
[pfcWatchdogPortTimer-Values] **	*PfcWatchdogPortTimer-Model* see 2.1634.1.2	Configured per-port timer values for PFC Watchdog. **Since: 4.20.0
schedulerCompensation-Supported **	*boolean*	Scheduler compensation supported. **Since: 4.26.2
trustMode **	*enum: untrusted | cos | invalid | dscp*	Qos Trustmode of interface. **Since: 4.17.0
txQueueProfiles	*table(numeric string) = TxQueueProfileModel* see 2.1634.1.3	A mapping between Transmit queue id and Transmit Queue configuration.

2.1634.1.1 PfcProfileModel

dynamicLoadBalancing	*boolean*	Dynamic load balancing.

2.1634.1.2 PfcWatchdogPortTimerModel

forcedRecovery	*boolean*	Force recover any stuck queue(s) after the recoveryTimer duration, irrespective of whether PFC frames are being received or not.
pollingInterval	*floating point number*	Configure the interval at which the watchdog should poll the queue(s).
recoveryTime	*floating point number*	Configure recovery-time after which stuck queue should recover and start forwarding.
timeout	*floating point number*	Configure timeout after which port should be errdisabled or should start dropping on congested priorities or just notify about the event.

2.1634.1.3 TxQueueProfileModel

txQueueModel	*TxQueue* see 2.1634.1.3.1	Transmit Queue model.

2.1634.1.3.1 TxQueue

[dpThresholds] **	*table(numeric string) = DropPrecedenceThresholdsModel* see 2.1634.1.3.1.1	Tail drop thresholds per Drop Precedence. **Since: 4.22.0
[dpWred] **	*table(numeric string) = WredParametersModel* see 2.1624.1.1.1	Transmit Queue Dp WRED Parameters.Not Applicable if Transmit Queue Dp WRED is disable. **Since: 4.28.1
[ecn]	*EcnParametersModel* see 2.1614.2	Transmit Queue ECN Parameters. Not Applicable if Transmit Queue ECN is disabled.
[ecnDelay] **	*EcnDelayConfigurationModel* see 2.1634.1.3.1.2	Transmit Queue Delay based ECN Parameters. Not Applicable if Transmit Queue delay ECN is disabled. **Since: 4.17.0
ecnMarkProbSupported **	*boolean*	ECN Mark Probability Supported,. **Since: 4.21.3
ecnWeightSupported **	*boolean*	ECN Weight Supported. **Since: 4.21.3
[guaranteedBw]	*GuaranteedBwModel* see 2.1617.1.1.2	Configured guaranteed bandwidth.
[latencyThreshold] **	*LatencyThresholdModel* see 2.1634.1.3.1.3	Configured latency threshold. **Since: 4.23.0
[noPriority] **	*boolean*	Transmit Queue priority. **Since: **currentRelease**
[nonEct] **	*EcnParametersModel* see 2.1614.2	Transmit Queue NON-ECT Parameters. Not Applicable if Transmit Queue NON-ECT is disabled. **Since: 4.20.5
nonEctWeightSupported **	*boolean*	NON-ECT Weight Supported. **Since: 4.21.3

[shapeRate]	*ShapeRateModel* see 2.477.1.1.1.2	Configured shape rate.
txQueue	*integer*	Transmit Queue id.
[txQueueType] **	*enum: uc-tx-queue | mc-tx-queue | tx-queue*	Type of Transmit Queue. It is tx-queue for platforms that don't support multicast and unicast queues. **Since: 4.17.0
[weightConfig] **	*integer*	Configured weight. **Since: **currentRelease**
[wred] **	*WredParametersModel* see 2.1624.1.1.1	Transmit Queue WRED Parameters. Not Applicable if Transmit Queue WRED is disabled. **Since: 4.17.0
wredWeightSupported **	*boolean*	WRED Weight Supported. **Since: 4.21.3
[wrrBandwidthPercent]	*integer*	Configured WRR bandwidth in percent.

2.1634.1.3.1.1 DropPrecedenceThresholdsModel

percent	*integer*	Tail drop-thresholds in percent of correspondingDrop Precedence 0 threshold value.

2.1634.1.3.1.2 EcnDelayConfigurationModel

Could also be an instance of:
EcnDelayParametersModel 2.1614.1

[configuredThreshold]	*floating point number*	Configured Threshold value in microseconds.
[source]	*enum: global | none | local*	Configuration source.

2.1634.1.3.1.3 LatencyThresholdModel

threshold	*integer*	Configured latency threshold.
unit	*string*	Latency threshold unit.

2.1635 show qos profile summary [...]

qosProfileApplication	*table(string) = QosProfileConfiguredIntfs* see 2.1635.1	A mapping between QoS Profile name and configured interfaces.

2.1635.1 QosProfileConfiguredIntfs

intfs	*list of Interface name*	Interface name.

2.1636 show qos random-detect ecn [...]

Could also be an instance of:
GlobalAllEcnModel 2.1614

delayEcnSupported **	*boolean*	Global Delay based ECN supported. **Since: 4.17.0
[globalEcnDelayParameters] **	*EcnDelayParametersModel* see 2.1614.1	Global delay based ECN Parameters. Not applicable if Global delay based ECN is disabled. **Since: 4.17.0
[globalEcnParameters]	*EcnParametersModel* see 2.1614.2	Global length based ECN Parameters. Not applicable if Global length based ECN is disabled.
globalEcnSupported **	*boolean*	Global Buffer ECN supported. **Since: 4.17.0
globalEcnWeightSupported **	*boolean*	Global ECN Supported. **Since: 4.21.3

see 2.1614

2.1637 show qos scheduling group [...]

interfaces	*table(Interface name) = IntfSchedulingGroupWithMembersModel* see 2.1637.1	A mapping of interface to its QoS scheduling groups.

2.1637.1 IntfSchedulingGroupWithMembersModel

schedulingGroups	*table(string) = SchedulingGroupWithMembersModel* see 2.1617.1.1.3.3	SchedulingGroups configured for the interface, indexed by their name.

2.1638 show qos scheduling hierarchy [...]

interfaces	*table(Interface name) = QosSchedulingHierarchyModel* see 2.1638.1	A mapping of sub interface to its QoS scheduling hierarchy.

2.1638.1 QosSchedulingHierarchyModel

parentInterface	*HierarchyModel* see 2.1617.1.1.3.2	Scheduling information for the parent interface.
[schedulingGroup]	*HierarchyModel* see 2.1617.1.1.3.2	Scheduling group containing the subinterface.
subinterface	*HierarchyModel* see 2.1617.1.1.3.2	Scheduling information for the subinterface.
[txQueues]	*list of HierarchyModel* see 2.1617.1.1.3.2	List of transmit queues on the subinterface.

2.1639 show queue-monitor length (all | global-buffer | statistics)

bytesPerTxmpSegment	*integer*	Number of bytes per txmp segment.
entryList	*list of CongestionRecordModel* see 2.1639.1	List of the congestion records.
globalHitCount	*integer*	Number of times the global High threshold is hit.
lanzEnabled	*boolean*	Queue length monitoring enabled.
[platformName]	*enum: Strata | Mako | Xp | Sand | Barefoot*	Forwarding plane name.
reportTime	*floating point number*	UTC time at which the congestion report is generated.

2.1639.1 CongestionRecordModel

Could also be an instance of:
BarefootCongestionRecord 2.1639.2
BarefootStatisticsRecord 2.1639.3
MakoCongestionRecord 2.1639.4
MakoStatisticsRecord 2.1639.5
SandCongestionRecord 2.1639.6
SandLatencyRecord 2.1639.7
StrataCommonRecord 2.1639.8
StrataCongestionRecord 2.1639.9
StrataDropsRecord 2.1639.10
StrataLatencyRecord 2.1639.11
StrataStatisticsRecord 2.1639.12
XpCongestionRecord 2.1639.13
XpStatisticsRecord 2.1639.14

2.1639.2 BarefootCongestionRecord

duration	*integer*	Congestion duration in microseconds.
entryTime	*floating point number*	UTC time in seconds at which the congestion record occurred.
interface	*interface name*	Interface on which the congestion event occurred.
queueLength	*integer*	Queue length in segments.
txQueue	*integer*	TX-queue associated with the congestion.

2.1639.3 BarefootStatisticsRecord

interface	*interface name*	Interface on which the congestion events occurred.
numCongestionEvents	*integer*	Number of times congestion started on the interface per TX-Queue.
txQueue	*integer*	Traffic class associated with the congestion events.

2.1639.4 MakoCongestionRecord

bufferType	*enum: E | CM | I*	Congestion buffer type, I-Ingress, E-Egress, CM-Core memory.
durationUsecs	*integer*	congestion duration in microseconds.
entryTime	*floating point number*	UTC time in seconds at which the congestion record occurred.
entryType	*enum: E | S*	Congestion entry type, S-Start, E-End.
interface	*interface name*	Interface on which the congestion event occurred.

2.1639.5 MakoStatisticsRecord

bufferType	*integer*	Buffer associated with the congestion event.
interface	*interface name*	Interface on which the congestion events occurred.
numCongestionEvents	*integer*	Number of times congestion started on the interface per buffer type.

2.1639.6 SandCongestionRecord

duration	*integer*	congestion duration in seconds.
[durationUsecs]	*integer*	congestion duration in microseconds.
entryTime	*floating point number*	UTC time in seconds at which the congestion record occurred.
[entryTimeUsecs]	*integer*	UTC time in microseconds at which the congestion record occurred.
entryType	*enum: E | P | S | U*	Congestion entry type, S-Start, U-Update, E-End, P-Polling.
globalProtectionModeEnabled **	*boolean*	Global protection mode is enabled. **Since: 4.20.5
ingressPortSet	*list of Interface name*	Ingress port-set corresponding to the interfacewith the congestion.
interface	*string*	Interface on which the congestion event occurred. We monitor congestion for the front panel ports as well as the cpu port.
queueLength	*integer*	Queue length in bytes.
trafficClass	*integer*	Traffic-class associated with the congestion, if applicable. Number of traffic classes supported are 8(0 - 7).

2.1639.7 SandLatencyRecord

entryTime	*floating point number*	Time at which the TX latency is recorded.
globalProtectionModeEnabled	*boolean*	Global protection mode is enabled.
ingressPortSet	*list of Interface name*	Ingress port-set corresponding to the interface with the tx-latency.
interface	*interface name*	Destination interface.
trafficClass	*integer*	Traffic class associated with the tx delay. Number of traffic classes supported are 8(0 - 7).

txLatency	*floating point number*	Tx-delay information in usecs.

2.1639.8 StrataCommonRecord

[cardSlot]	*integer*	Card slot associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records.
[chipIndex]	*integer*	Chip index associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records.
interface	*interface name*	Interface on which the congestions events occurred.
trafficClass	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 22 (0-11 for unicast, 12-21 for multicast).

2.1639.9 StrataCongestionRecord

[cardSlot] **	*integer*	Card slot associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
[chipIndex] **	*integer*	Chip index associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
congestionDuration	*integer*	Congestion duration associated with the congestion event (usecs). Only applicable with entries of type E. Value -1 represents an undeterministic value.
entryTime	*floating point number*	Time at which the congestion entry is generated.
entryType	*enum: E | S | U*	Congestion entry type, S-Start, U-Update, E-End.
[fabricPeerIntf] **	*string*	Peer interface for the congested fabric interface. **Since: 4.17.0
interface	*interface name*	Interface on which the congestions events occurred.
maxQueueLengthTime	*integer*	Time at which Max Queue length occurred relative to congestion start (usecs). Only applicable with entries of type E. Value of -1 represents an undeterministic value.

queueLength	*integer*	Queue length in segments, Segment size for S, U, and E type records varies per chip model.
trafficClass	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 22 (0-11 for unicast, 12-21 for multicast).
txDrops	*integer*	Number of packet drops seen on the interface due to congestion.
txLatency	*floating point number*	Tx-delay information per interface (usecs).

2.1639.10 StrataDropsRecord

[cardSlot] **	*integer*	Card slot associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
[chipIndex] **	*integer*	Chip index associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
entryTime	*floating point number*	UTC time at which congestion drops are seen.
interface	*interface name*	Interface on which the congestions events occurred.
trafficClass	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 22 (0-11 for unicast, 12-21 for multicast).
txDrops	*integer*	Number of packet drops seen on the interface due to congestion.

2.1639.11 StrataLatencyRecord

[cardSlot] **	*integer*	Card slot associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
[chipIndex] **	*integer*	Chip index associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
entryTime	*floating point number*	Time at which the tx latency is recorded.
interface	*interface name*	Interface on which the congestions events occurred.

trafficClass	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 22 (0-11 for unicast, 12-21 for multicast).
txLatency	*floating point number*	Tx-delay information in usecs.

2.1639.12 StrataStatisticsRecord

[cardSlot] **	*integer*	Card slot associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
[chipIndex] **	*integer*	Chip index associated with the congestion event.Needed for distinguishing congestion records from CPU ports on multi-switch chips. Only valid for CPU congestion records. **Since: 4.17.0
interface	*interface name*	Interface on which the congestions events occurred.
numCongestionEvents	*integer*	Number of times congestion started on the interface per traffic-class.
trafficClass	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 22 (0-11 for unicast, 12-21 for multicast).

2.1639.13 XpCongestionRecord

congestionDuration **	*integer*	congestion duration in microseconds. **Since: 4.20.0
entryTime	*floating point number*	UTC time in seconds at which the congestion record occurred.
entryType	*enum: E | S | U*	Congestion entry type, S-Start, U-Update, E-End.
interface	*interface name*	Interface on which the congestion event occurred.
maxQueueLengthTime	*integer*	Time at which Max Queue length occurred relative to congestion start (usecs). Only applicable with entries of type E. Value of -1 represents an undeterministic value.
queueLength	*integer*	Queue length in pages of 256 bytes.
trafficClass	*integer*	Traffic-class associated with the congestion.

2.1639.14 XpStatisticsRecord

interface	*string*	Interface on which the congestion events occurred.
numCongestionEvents	*integer*	Number of times congestion started on the interface per traffic-class.

`trafficClass`	*integer*	Traffic class associated with the congestion events. Number of traffic classes supported are 8(0 - 7).

2.1640 show queue-monitor length [...]

see 2.1639

2.1641 show queue-monitor length csv [...]

see 2.1639

2.1642 show queue-monitor length status

bytesPerRxmpSegment	*integer*	Rxmp segment size in bytes.
bytesPerTxmpSegment	*integer*	Txmp segment size in bytes.
globalBufferEnabled	*boolean*	Global buffer monitoring enabled.
globalBufferSupported	*boolean*	Global buffer monitoring supported.
globalHighThreshold	*integer*	High thresold on global buffer.
globalLogInterval	*floating point number*	Syslog interval for global buffer monitoring.
globalLowThreshold	*integer*	Low threshold on global buffer.
intfStatusAll	*table(Interface name) = InterfaceStatusModel* see 2.1642.1	List of Interfaces.
lanzEnabled	*boolean*	Queue length monitoring enabled.
[lanzMode]	*enum: polling | notifying*	Configured queue length monitoring mode.
[lanzSliceModes] **	*table(string) = string*	Mapping of Slice name to Queue length monitoring status. **Since: 4.20.0
[latencyCapableLinecards] **	*table(string) = boolean*	Linecards capable of measuring ingresslatency. **Since: 4.22.1
logInterval	*floating point number*	Syslog interval in seconds.
lowThresholdSupported	*boolean*	lower threshold supported.
[makoShowQMonLenStatusModel] **	*MakoQueueMonitorLengthStatusModel* see 2.1642.2	Mako queue monitor length status. **Since: 4.29.0
maxQueueLength	*integer*	Maximum queue length.
mirrorDestIntf	*list of string*	Mirror destination interfaces.
mirroringSupported **	*boolean*	Mirroring supported. **Since: 4.15.2
numTotalSegments	*integer*	Total buffers in segments.
packetSamplingEnabled	*boolean*	Queue-Monitor Length Mirroring Enabled.
[platformName] **	*enum: Strata | Mako | Xp | Sand | Barefoot*	Forwarding plane name. **Since: 4.29.0
ptpSupported **	*boolean*	PTP is supported on current system. **Since: 4.22.1
qLenInterval	*integer*	Queue length interval.
queueLenUnit	*enum: bytes | segments*	Queue length specified in segments or bytes.
txLatencyEnabled **	*boolean*	Latency measurement enabled. **Since: 4.22.1
txLatencyEnablingSupported **	*boolean*	Latency measurement enabling supported. **Since: 4.28.1
updateIntervalSupported	*boolean*	Update interval supported.
[updateIntervalSupportedCards] **	*table(string) = boolean*	List of cards types that support update interval. **Since: 4.23.0

2.1642.1 InterfaceStatusModel

highThreshold	*integer*	High threshold configured on the interface.
[inactiveWarning] **	*InactiveWarningModel* see 2.1642.1.1	Thresholds inactive if greater than the interface's maximum queue length. **Since: 4.20.0
lanzDisabled	*boolean*	Queue length monitoring disabled status on the interface.
lowThreshold	*integer*	Low threshold configured on the interface.
mirroringEnabled	*boolean*	Queue-monitor length mirroring enabled on the interface.

2.1642.1.1 InactiveWarningModel

maxQueueLength	*integer*	Maximum VOQ tail drop threshold.

2.1642.2 MakoQueueMonitorLengthStatusModel

[corememHighThreshold]	*integer*	High threshold on core memory buffer in bytes.
[corememLowThreshold]	*integer*	Low threshold on core memory buffer in bytes.
[corememMemorySize]	*integer*	Memory size of core memory buffer in bytes.
egressHighThreshold	*integer*	High threshold on egress buffer in bytes.
egressLowThreshold	*integer*	Low threshold on egress buffer in bytes.
egressMemorySize	*integer*	Memory size of egress buffer in bytes.
ingressHighThreshold	*integer*	High threshold on ingress buffer in bytes.
ingressLowThreshold	*integer*	Low threshold on ingress buffer in bytes.
ingressMemorySize	*integer*	Memory size of ingress buffer in bytes.
lanzEnabled	*boolean*	Queue length monitoring enabled.

2.1643 show queue-monitor streaming ((ip access-list) | (ipv6 access-list)) [...]

see 2.53

2.1644 show queue-monitor streaming clients

clientHostnames	*list of string*	Hostnames of clients connected to Lanz Streaming.
lanzEnabled	*boolean*	Queue length monitoring enabled.
streamingEnabled	*boolean*	Queue length streaming enabled.

2.1645 show radius

[groups]	*table(string) = ServerGroup* see 2.1645.2	Mapping of group name to group information.
lastCounterClearTime	*floating point number*	UTC time when the counters were last cleared.
radiusServers	*list of RadiusStats* see 2.1645.1	Mapping of server IP address or hostname to server information.
srcIntf	*table(string) = Interface name*	Configured source-interface name.

2.1645.1 RadiusStats

acctInterimUpdatesSent **	*integer*	Number of Accounting Interim Update messages sent to RADIUS server. **Since: 4.29.1
acctStartsSent **	*integer*	Number of Accounting Start messages sent to RADIUS server. **Since: 4.29.1
acctStopsSent **	*integer*	Number of Accounting Stop messages sent to RADIUS server. **Since: 4.29.1
badResponses	*integer*	Number of bad responses from the RADIUS server.
coaAckSent **	*integer*	Number of CoA Ack responses sent to RADIUS server. **Since: 4.22.1
coaNakSent **	*integer*	Number of CoA Nak responses sent to RADIUS server. **Since: 4.22.1
coaRequestsReceived **	*integer*	Number of CoA requests received from RADIUS server. **Since: 4.22.1
connectionErrors **	*integer*	Number of connection errors to the RADIUS server. **Since: 4.25.1
dmAckSent **	*integer*	Number of DM Ack responses sent to RADIUS server. **Since: 4.22.1
dmNakSent **	*integer*	Number of DM Nak responses sent to RADIUS server. **Since: 4.22.1
dmRequestsReceived **	*integer*	Number of DM requests received from RADIUS server. **Since: 4.22.1
dnsErrors	*integer*	Number of DNS errors by the RADIUS server.
messagesReceived	*integer*	Number of packets received from the RADIUS server.
messagesSent	*integer*	Number of packets sent to the RADIUS server.
requestsAccepted	*integer*	Number of requests accepted by the RADIUS server.

requestsRejected	*integer*	Number of requests rejected by the RADIUS server.
requestsRetransmitted	*integer*	Number of requests retransmitted by the RADIUS server.
requestsTimeout	*integer*	Number of requests to the RADIUS server timed out.
serverInfo	*ServerInfo* see 2.1645.1.1	Configured server information.

2.1645.1.1 ServerInfo

[acctport]	*integer*	Specifies server accounting port number.
authport	*integer*	Specifies server authorization port number.
[dot1xDynAuthEnabled] **	*boolean*	Specifies if dynamic authorization is enabled. **Since: 4.30.1
[dynAuthPort] **	*integer*	Specifies server dynamic authorization port number. **Since: 4.22.1
hostname	*string*	IP address or hostname of the RADIUS server.
[sslProfile] **	*string*	SSL profile for RADIUS server. **Since: 4.26.2
[tlsport] **	*integer*	RADIUS server TLS port number. **Since: 4.26.2
[vrf] **	*string*	The VRF over which this server is reachable. **Since: 4.25.1

2.1645.2 ServerGroup

members	*list of ServerInfo* see 2.1645.1.1	Mapping of server IP address or hostname to group information.
serverGroup	*string*	Configured server-group.

2.1646 show recirc-channel [...]

see 2.325

2.1647 show recirc-channel limits

recircChannelLimits	*table(string) = LagGroupLimits* see 2.1565.1	Map a LAG Group name to the ports it applies to, and the hardware lag limits applicable to them.

2.1648 show recirc-channel traffic [...]

see 2.1567

2.1649 show redundancy file-replication

see 2.268.1.1

2.1650 show redundancy states

[agentsNotReady]	*list of string*	Agents not ready for SSO switchover.
allAgentSsoReady **	*boolean*	Agents not ready for SSO switchover. **Since: 4.26.0
communicationDesc	*enum: Up | Down*	Description of communication link.
configuredProtocol	*enum: sso | unknownProtocol | rpr | simplex*	Configured redundancy protocol.
globalStageCompletion-Timeout **	*floating point number*	switchover completion timeout in seconds. **Since: 4.17.0
globalStageCompletion-TimeoutDefault **	*floating point number*	switchover completion timeout default in seconds. **Since: 4.17.0
lastRedundancyModeChange-Reason	*string*	Reason for last redundancy mode change.
[lastRedundancyModeChange-Time]	*floating point number*	Time of last redundancy mode change in UTC.
myMode	*enum: disabled | active | standby | unknown*	Redundancy mode of the supervisor.
operationalProtocol	*enum: sso | unknownProtocol | rpr | simplex*	Operational redundancy protocol.
[pcieModulesNotReady]	*list of string*	PCIE modules not ready for SSO switchover.
[pcieSsoReady]	*boolean*	Is PCIE switch ready for switchover.
peerMode	*enum: disabled | active | standby | unknown*	Redundancy mode of the peer supervisor.
peerState	*enum: notInserted | unknownPeerState | inserted | poweringOn | poweredOff*	Standby supervisor state.
slotId	*integer*	Slot id of the supervisor.
switchoverReady	*boolean*	Is switch ready for switchover.
switchoverReadyTimeElapsed **	*boolean*	Has the mandatory wait time after switchover elapsed. **Since: 4.26.2
unitDesc	*enum: Primary | Secondary*	Description of supervisor.

2.1651 show redundancy status

see 2.1650

2.1652 show redundancy switchover sso

[lastSwitchoverStatus]	*enum: In Progress | Completed*	Status of last switchover.
[lastSwitchoverTime]	*floating point number*	Time of last switchover.
[logs]	*list of SwitchoverLog* see 2.1652.1	List of switchover logs.

switchoverCount	*integer*	Number of switchovers since reload.

2.1652.1 SwitchoverLog

msg	*string*	Log message.
time	*floating point number*	Time of the log relative to onset of switchover.

2.1653 show redundancy time-sync

currentMonoTimeDelta	*floating point number*	Time difference of current Monotonic time (seconds).
failCount	*integer*	Number of times that Monotonic time failed to update.
[lastErrorUpdating]	*string*	Reason for last error while updating Monotonic time.
lastMonoTimeUpdate	*floating point number*	Time of last update to Monotonic time sync.

2.1654 show reload

[allSupervisors]	*boolean*	All supervisors will be reloaded.
[reason]	*string*	Reason why a reload has been scheduled.
scheduled	*boolean*	A future reload is scheduled.
[scheduledTime]	*floating point number*	Time of the future reload (UTC).

2.1655 show reload cause [...]

[full]	*boolean*	Partial or full log.
[kernelCrashData] **	*list of string*	Kernel Panic EEPROM crash data. **Since: 4.24.2
resetCauses	*list of ReloadCause* see 2.1655.1	List of reload causes.

2.1655.1 ReloadCause

[debugInfo]	*list of string*	Information from debug file.
[debugInfoIsDir]	*boolean*	True if debug info is a directory.
description	*string*	Description of the reset cause.
[recommendedAction]	*string*	Action that is recommended.
[timestamp]	*floating point number*	Time of the reload.

2.1656 show reload cause history

resetHistory	*table(numeric string) = ReloadCauseList* see 2.1656.1	History of reload causes.

2.1656.1 ReloadCauseList

resetCauses	*list of ReloadCause* see 2.1656.1.1	List of reload causes.

2.1656.1.1 ReloadCause

[debugInfo]	*list of string*	Information from debug file.
[debugInfoIsDir]	*boolean*	True if debug info is a directory.
description	*string*	Description of the reset cause.
[recommendedAction]	*string*	Action that is recommended.
[timestamp]	*floating point number*	Time of the reload.

2.1657 show reload console logs [...]

consoleLog	*string*	Console log.

2.1658 show reload console status

blackboxEnabled	*boolean*	Console logging enabled.
blackboxManufacturer **	*string*	FRAM Manufacturer. **Since: 4.30.1
blackboxProductId **	*integer*	FRAM Product ID. **Since: 4.30.1
blackboxSupported	*boolean*	Console logging supported.
consoleLogFiles	*table(numeric string) = ConsoleLogFile* see 2.1658.1	Available ConsoleLog files.

2.1658.1 ConsoleLogFile

dumpTime	*string*	ConsoleLog save time.
fileName	*string*	ConsoleLog file name.
idVal	*integer*	ConsoleLog file id.

2.1659 show reload fast-boot

blockingList	*list of ReloadHitless-BlockingReason* see 2.1659.1	List blocking configuration for 'reload fast-boot'.
warningList	*list of ReloadHitless-WarningReason* see 2.1659.2	List of warnings for 'reload fast-boot'.

2.1659.1 ReloadHitlessBlockingReason

reason	*enum: stpMode* | *lacpRateFastLocalConfigured* | *portRole* | *maintenanceModeConfigured* | *mlagStateUnstable* | *transmitActive* | *vrrpEnabled* | *tunnelIntfConfigured* | *openFlowEnabled* | *lagModeInconsistent* | *resilientECMPv6* | *IPv6uRPF* | *IPv4uRPF* | *mlagNotEstablished* | *mstInstances* | *edgePort* | *directFlowEnabled* | *pbrConfigured* | *stpBpduguard* | *lacpRateFastRemoteConfigured* | *vxlanConfigured* | *resilientECMPv4*	IPv4uRPF : IPv4 uRPF is in ErrorDisabled state IPv6uRPF : IPv6 uRPF is in ErrorDisabled state directFlowEnabled : Directflow is enabled vrrpEnabled : VRRP is enabled resilientECMPv4 : Resilient ECMP prefix is configured for IPv4 resilientECMPv6 : Resilient ECMP prefix is configured for IPv6 lacpRateFastLocalConfigured : lacp fast rate configured locally lacpRateFastRemoteConfigured : lacp fast rate configured remotely openFlowEnabled : Openflow is enabled pbrConfigured : Pbr is configured stpMode : Spanning-tree mode is not supported edgePort : Spanning-tree portfast is not enabled for one or more ports portRole : One or more non-edge ports are in spanning-tree designated/backup role transmitActive : Transmit active is enabled for one or more ports mstInstances : Spanning-tree is enabled with more than one MST instance stpBpduguard : Spanning-tree BPDU guard is not enabled for one or more ports maintenanceModeConfigured : Maintenance Mode is configured lagModeInconsistent : Lag Modes differ between hardware and configuration mlagStateUnstable : Mlag state is neither primary nor secondary mlagNotEstablished : Mlag peer negotiation is not done yet vxlanConfigured : Vxlan is configured tunnelIntfConfigured : Tunnel Interface is configured.

2.1659.2 ReloadHitlessWarningReason

reason	*enum: igmpSnoopingEnabled* | *reloadDelayConfigured* | *mlagPortsActivePartial* | *pfcWatchdogEnabled* | *ospf3NoGrace* | *runtimeErrdisabledIntfs* | *stpErrDisabled* | *ptpEnabled* | *mlagVersionCompatibility* | *stpState* | *pimHoldTime* | *pimEnabled* | *bgpNoGrace* | *ripEnabled* | *macSecFipsRestrictions* | *mlagConfigured* | *pimBidirEnabled* | *isisNoGrace* | *ospfNoGrace* | *varpEnabled*	bgpNoGrace : BGP is configured without graceful restart ospfNoGrace : OSPF is configured without graceful restart isisNoGrace : IS-IS is configured without graceful restart ospf3NoGrace : OSPF3 is configured without graceful restart pimEnabled : PIM is enabled ripEnabled : RIP is enabled varpEnabled : VARP is enabled igmpSnoopingEnabled : IGMP Snooping is enabled stpErrDisabled : One or more ports is errDisabled due to BPDU guard stpState : STP is not currently restartable mlagConfigured : Mlag is configured runtimeErrdisabledIntfs : Runtime errdisabled interface(s) found. Please note that they will be re-enabled after hitless reload. If same conditions persist, interface(s) will be errdisabled again after the reload. ptpEnabled : PTP is enabled. PTP slaves will lose connection to the grandmaster and lose time sync. pfcWatchdogEnabled : PFC Watchdog is enabled. The current PFC Watchdog state may be lost on hitless reload. Monitoring will restart after the reload. mlagVersionCompatibility : If you are performing an upgrade, and the Release Notes for the new version of EOS indicate that MLAG is not backwards-compatible with the currently installed version (None), the upgrade will result in packet loss. mlagPortsActivePartial : reloadDelayConfigured : Mlag is configured with non-zero mlag/non-mlag reload-delay. This may result in packet loss pimBidirEnabled : PIM bidirectional is enabled pimHoldTime : Pim interface has join/prune holdtime that is less than two times default holdtime. This may result in packet loss macSecFipsRestrictions : MACsec FIPS restrictions are enabled. This may result in an extended traffic outage to perform FIPS POST during the reload.

2.1660 show reload hitless

blockingList	*list of ReloadHitlessBlockingReason* see 2.1659.1	List blocking configuration for 'reload hitless'.
warningList	*list of ReloadHitlessWarningReason* see 2.1659.2	List of warnings for 'reload hitless'.

2.1661 show rib loop (ip | ipv6) route [...]

loopedRoutes	*table(IPv4 or IPv6 address prefix) = LoopedRoute* see 2.1661.1	Map of routes involved in recursive route resolution loops.

2.1661.1 LoopedRoute

allNextHopsPartOfLoop	*boolean*	Set to true if all next-hops of this route are part of recursive route resolution loops.
protocol	*string*	The protocol for the looped route.

2.1662 show rib multicast route (ip | ipv6) [...]

revision 4

see 2.658

2.1663 show rib next-hop (ip | ipv6) [...]

revision 2

ribNextHopsByProtocol	*table(string) = RibNextHop* see 2.1663.1	Map of next-hop vias for each routing protocol.

2.1663.1 RibNextHop

[recursiveNextHops]	*list of RecursiveNextHop* see 2.658.1.1.1.1	List of next-hop routers through which routes in the RIB are reachable.
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	List of immediate next-hop routers through which routes in the RIB are reachable.
[ribNextHop]	*RibNextHopRibNextHop* see 2.1663.1.1	RIB nexthop.

2.1663.1.1 RibNextHopRibNextHop

[recursiveNextHops]	*list of RecursiveNextHop* see 2.658.1.1.1.1	Map of next-hop routers through which routes in the RIB are reachable.
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	List of immediate next-hop routers through which routes in the RIB are reachable.

2.1664 show rib next-hop (ip | ipv6) dependency

[ribNhDependencyByVrf]	*table(string) = RecursiveNhDependencyByProtocol* see 2.1664.1	Next-hop dependencies for all vrfs.

2.1664.1 RecursiveNhDependencyByProtocol

[ribNhDependencyByProtocol]	*table(string) = RecursiveNhDependencies* see 2.1664.1.1	Next-hop dependencies for all protocols in this vrf.

2.1664.1.1 RecursiveNhDependencies

[nextHops]	*list of RecursiveNhDependency* see 2.1664.1.1.1	List of next-hops and their dependencies.

2.1664.1.1.1 RecursiveNhDependency

[loopedNextHop]	*boolean*	Set to true if this next-hop is part of a recursive route resolution loop, false otherwise.
[nextHop]	*IP address*	Next hop router address.
[recursiveNextHops]	*list of RecursiveNhDependency* see 2.1664.1.1.1	The list of LPM next-hop routers through which this recursive next-hop is reachable
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	The list of immediate next-hop routers through which this recursive next-hop is reachable.
[resolvingRoute]	*IP address prefix*	Covering route for this next-hop.
[routeType]	*enum: staticConfig | ospf | routingProtocols | connected | routeInput | bgp | ospf3 | rip | isis | dynamicPolicy | vrfLeak | reserved*	Protocol through which resolving route was learned.
[viaId]	*integer*	ID of Via.
[vrfName]	*string*	Name of VRF from which this next-hop was requested.

2.1665 show rib next-hop resolution route (ipv4 | ipv6) [...]

revision 2

[routeMap]	*string*	RIB recursive resolution policy.
[routes]	*table(IPv4 or IPv6 address prefix) = ResolutionRoute* see 2.1665.1	Map of routes to their recursive resolution status.

2.1665.1 ResolutionRoute

metric	*integer*	The routing protocol internal metric (cost) for this route.
metricType	*enum: MED | metric | AIGP*	The type of metric.
preference	*integer*	The administrative distance of this route.

protocol	*enum: staticConfig* | *ospf* | *routingProtocols* | *connected* | *routeInput-RouteCacheConfig* | *routeInput* | *staticRouteCacheConfig* | *bgp* | *ospf3* | *rip* | *isis* | *dynamicPolicy* | *cbf* | *gribi* | *vrfLeak* | *reserved*	The routing protocol for this route.
[recursiveNextHops]	*list of RecursiveNextHop* see 2.658.1.1.1.1	List of next-hop routers through which routes in the RIB are reachable.
resolvability	*enum: reject* | *uninitialized* | *accept*	The resolvability of this route.
[resolvedNextHops]	*list of ResolvedNextHop* see 2.658.1.1.1.4	List of immediate next-hop routers through which this route is reachable.

2.1666 show rib ready [...]

vrfs	*table(string) = RibReadyModel* see 2.1666.1	Per VRF Rib ready summary.

2.1666.1 RibReadyModel

[flags] **	*RibReadyFlags* see 2.1666.1.2	Various checkpoints along the path to declare Fib readyin ribd protocol model. **Since: 4.21.0
gracefullyRestarting	*boolean*	The vrf is gracefully restarting or not.
[protocols]	*table(string) = ProtoReadyModel* see 2.1666.1.1	Current state of the protocols taking part in the Rib ready process in multi-agent protocol model, key being protocol name.
[routes] **	*table(string) = RouteReady* see 2.1666.1.4	Current state of the route types considered part of the Rib ready process in ribd protocol model, key being route type. **Since: 4.21.0
startTime	*floating point number*	UTC time at which Rib ready verification was started.
[states]	*RibReadyStateModel* see 2.1666.1.3	Various states that occur during the Rib ready verification in the multi-agent protocol model.

2.1666.1.1 ProtoReadyModel

configPending	*boolean*	Whether processing of protocol configuration is pending or not.
configured	*boolean*	Whether the protocol is configured.
ready	*boolean*	Whether the protocol has converged.
[readyTime]	*floating point number*	UTC timestamp when this protocol became ready.
[timeTaken]	*floating point number*	Time elapsed from start until this protocol becomes ready.

2.1666.1.2 RibReadyFlags

fibReady	*boolean*	Whether the routes have been synced to FIB.
recursiveResolution	*boolean*	Whether the static routes have been recursively resolved.
ribReady	*boolean*	Whether all protocols have converged.
[timeTakenForFibReady]	*floating point number*	Time elapsed in seconds from start until FIB became ready.
[timeTakenForRecursive-Resolution]	*floating point number*	Time elapsed in seconds from start until static routes were recursively resolved.
[timeTakenForRibReady]	*floating point number*	Time elapsed in seconds from start until all protocols converged.
[timeTakenForV4RouteSync]	*floating point number*	Time elapsed in seconds from start until v4 routes were synced.
[timeTakenForV6RouteSync]	*floating point number*	Time elapsed in seconds from start until v4 routes were synced.

v4RouteSync	*boolean*	Whether v4Routes have been synced to Fib.
v6RouteSync	*boolean*	Whether v6 routes have been synced to Fib.

2.1666.1.3 RibReadyStateModel

fibReady	*boolean*	Whether the routes have been synced to FIB.
[fibReadyTime]	*floating point number*	UTC time at which all routes were synced to FIB.
igpReady	*boolean*	Whether IGPs have converged.
[igpReadyTime]	*floating point number*	UTC time at which IGPs converged.
recursiveResolution	*boolean*	Whether recursive resolution is complete.
[recursiveResolutionTime]	*floating point number*	UTC time at which recursive route resolution was complete.
ribReady	*boolean*	Whether all protocols have converged.
[ribReadyTime]	*floating point number*	UTC time at which all protocols converged.
[timeTakenForFibReady]	*floating point number*	Time elapsed in seconds from start until FIB became ready.
[timeTakenForIgpReady]	*floating point number*	Time elapsed in seconds from start until IGPs converged.
[timeTakenForRecursive-Resolution]	*floating point number*	Time elapsed in seconds from start until recursive resolution was complete.
[timeTakenForRibReady]	*floating point number*	Time elapsed in seconds from start until all protocols converged.

2.1666.1.4 RouteReady

configPending	*boolean*	Whether processing of route configuration is pending.
configured	*boolean*	Whether the route type is configured.
ready	*boolean*	Whether the route type has converged.
[timeTaken]	*floating point number*	Time elapsed from start until this route type became ready.

2.1667 show rib route (ip | ipv6) [...]

revision 4

see 2.658

2.1668 show rib route (ip | ipv6) internal [...]

revision 4

see 2.658

2.1669 show rib route (ip | ipv6) tag TAG [...]

revision 4

see 2.658

2.1670 show rib route summary [...]

ribRouteSummary	*table(string) = RibRouteSummaryForVrf* see 2.1670.1	Summary of routes in the RIB of each VRF.

2.1670.1 RibRouteSummaryForVrf

routeCountPerProtocol	*table(string) = integer*	Total number of routes for each routing protocol in the RIB.

2.1671 show rib route summary brief [...]

see 2.1670.1

2.1672 show rib system-tunneling-ldp-rib route ipv4 [...]

revision 2

see 2.691.1.1.3

2.1673 show role [...]

defaultRole	*string*	Default role.
roles	*table(string) = Rules* see 2.1673.1	Display the contents of built-in and user-defined roles.

2.1673.1 Rules

rules	*list of ShowRolesList* see 2.1673.1.1	A mapping of an authorization role name to its rules.

2.1673.1.1 ShowRolesList

cmdPermission	*enum: permit | deny*	If match is found against a rule, decision is based on that rule and all other rules followed are ignored. If no match is found, the command is denied by default.
cmdRegex	*string*	Regular expression that corresponds to one or more CLI commands.

[mode]	*string*	The regex of a cli mode name for which this rule applies, if absent, rule is valid in all modes.
sequenceNumber	*integer*	Sequence number designates a rule's placement in the role, value ranges from 1 to 256.

2.1674 show route (ipv4 | ipv6) counters [...]

vrfs	*table(string) = RouteCounters* see 2.1674.1	A mapping of VRF name to per prefix counters.

2.1674.1 RouteCounters

counters	*list of RouteCounterEntry* see 2.1674.1.1	List of prefix counters.
[uniqueBackup] **	*boolean*	Dedicated backup enabled in VRF. **Since: 4.29.1

2.1674.1.1 RouteCounterEntry

[backupByteCount] **	*integer*	Backup path byte count. **Since: 4.29.1
[backupPacketCount] **	*integer*	Backup path packet count. **Since: 4.29.1
byteCount	*integer*	Byte count.
packetCount	*integer*	Packet count.
prefix	*IP address prefix*	Route prefix.
[programmed] **	*boolean*	Counter is programmed in hardware. **Since: 4.29.1

2.1675 show route cache (vrf (VRF | all)) [...]

[cmdErr]	*string*	Error message.
[vrfs]	*table(string) = Prefix* see 2.1675.1	Mapping of VRF name to IP routes.

2.1675.1 Prefix

[routes]	*table(IPv4 or IPv6 address prefix) = PrefixData* see 2.1675.1.1	Mapping of prefix to its corresponding data.

2.1675.1.1 PrefixData

[inHw]	*boolean*	Prefix in hardware.
[nextHopId]	*integer*	Next-hop index.
[nextHopInfo]	*table(numeric string) = NextHopInfo* see 2.1675.1.1.1	Mapping of next-hop id to next-hop info.

2.1675.1.1.1 NextHopInfo

macAddress	*MAC address*	Overlay MAC address.
remoteVtep	*IP address*	Remote VTEP IP address.
underlayFecId	*integer*	Underlay FEC ID.
vlanId	*integer*	VLAN ID.
vni **	*integer*	VNI. **Since: 4.25.2
vxlanId **	*interface name*	VXLAN Interface. **Since: 4.25.2

2.1676 show route cache (vrf (VRF | all)) ipv6 [...]

see 2.1675

2.1677 show route cache summary [...]

[cmdErr]	*string*	Error Message.
[summary]	*table(string) = RouteCacheSummary* see 2.1677.1	Route cache summary for IPv4.
[summaryIPv6] **	*table(string) = RouteCacheSummary* see 2.1677.1	Route cache summary for IPv6. **Since: 4.25.2
total **	*RouteCacheSummary* see 2.1677.1	Total across all VRFs and address families. **Since: 4.25.2

2.1677.1 RouteCacheSummary

configuredRoutes	*integer*	Number of routes configured.
routesInHw	*integer*	Number of routes programmed in Hardware.
routesInSfe	*integer*	Number of routes programmed in SFE.
unprogrammedRoutes	*integer*	Number of unprogrammed routes.

2.1678 show route cache threshold

thresholdParams	*table(string) = ThresholdParams* see 2.1678.1	Threshold params for different levels.

2.1678.1 ThresholdParams

hwUtil	*integer*	Hardware space utilization in percentage for this threhold level to kick in.
idleTime	*integer*	Idle time limit for prefixes at this threhold level in seconds.

2.1679 show route-map [...]

revision 5

see 2.895.1.2

2.1680 show route-map config-sanity

revision 5

see 2.895.1.3

2.1681 show route-map expanded [...]

revision 5

[mapName]	*string*	Name of specific route map to render.
pruned	*boolean*	Avoid rerendering common subtrees.
routeMaps	*table(string) = ExpandedRouteMap* see 2.1681.1	Route maps by name.

2.1681.1 ExpandedRouteMap

[entries]	*table(numeric string) = ExpandedRouteMapEntry* see 2.1681.1.1	Route map entries by sequence number.

2.1681.1.1 ExpandedRouteMapEntry

[continueClause]	*boolean*	Continue to next clause.
[continueSeqno]	*integer*	Sequence number of next clause.
description	*list of string*	Route map entry description.
filterType	*enum: permit | deny*	Filter type.
matchRules	*MatchRules* see 2.127.1.3.2.1	Match rules.
setRules	*SetRules* see 2.127.1.3.2.8	Set rules.
[subRouteMap]	*SubRouteMap* see 2.895.1.2.1.2	Sub route map.

2.1682 show router igp topology [...]

revision 2

`vrfs`	*table(string) = VrfModel* see 2.1682.1	IGP topology database per VRF.

2.1682.1 VrfModel

`[isis]`	*IgpInstanceInfoModel* see 2.1682.1.1	IGP topology information for IS-IS.

2.1682.1.1 IgpInstanceInfoModel

`instances`	*table(numeric string) = IsisModel* see 2.1682.1.1.1	A mapping of the IGP instance ID to the IGP database.

2.1682.1.1.1 IsisModel

`[level1]`	*CommonProtocolModel* see 2.1682.1.1.1.1	IS-IS Level-1 topology information.
`[level2]`	*CommonProtocolModel* see 2.1682.1.1.1.1	IS-IS Level-2 topology information.

2.1682.1.1.1.1 CommonProtocolModel

`[ipv4]`	*table(string) = RouterModel* see 2.1682.1.1.1.2	IPv4 topology information per router ID.
`[ipv6]`	*table(string) = RouterModel* see 2.1682.1.1.1.2	IPv6 topology information per router ID.

2.1682.1.1.1.2 RouterModel

`[flexAlgo]` **	*list of integer*	List of all supported Flex Algos. **Since: 4.27.1
`[lans]`	*table(string) = NeighborModel* see 2.1682.1.1.1.3	Neighbor information per LAN neighbor.
`[p2ps]`	*table(string) = NeighborModel* see 2.1682.1.1.1.3	Neighbor information per P2P neighbor.
`[reachability]`	*table(IPv4 or IPv6 address prefix) = ReachabilityModel* see 2.1682.1.1.1.6	Reachability information per prefix.

2.1682.1.1.1.3 NeighborModel

`[flexAlgoAttributes]` **	*FlexAlgoAttributesModel* see 2.1682.1.1.1.4	flexAlgo related Attributes. **Since: 4.27.1
`[interfaceAddresses]`	*list of IP address*	Local interface IP addresses.

metric	*integer*	Link cost.
[neighborAddresses]	*list of IP address prefix*	Remote interface IP addresses.

2.1682.1.1.1.4 FlexAlgoAttributesModel

[administrativeGroup]	*integer*	Administrative Group of the Link.
[defaultTeMetric] **	*integer*	TE default metric. **Since: 4.30.1
[minimumDelayMetric]	*integer*	Minimum Delay Metric in microseconds.
[sharedRiskLinkGroups]	*list of SharedRiskLinkGroup* see 2.1682.1.1.1.5	List of Shared Risk Link Group.

2.1682.1.1.1.5 SharedRiskLinkGroup

srlgId	*integer*	Shared Risk Link Group Identifier.
[srlgName]	*string*	Shared Risk Link Group Name.

2.1682.1.1.1.6 ReachabilityModel

metric	*integer*	Link cost.

2.1683 show router rcf code [...]

rcfCodeUnits	*table(string) = string*	RCF code unit text keyed by code unit name.

2.1684 show router rcf errors [...]

active	*boolean*	Routing control functions are active.
compilationErrors	*list of string*	Routing control function running-config errors.

2.1685 show router rcf function location [...]

functions	*table(string) = FunctionLocationModel* see 2.1685.1	RCF function location information keyed by function name.

2.1685.1 FunctionLocationModel

codeUnit	*string*	Code unit that the function is stored in.
lineNum	*integer*	Line number of the start of the function within the code unit.

2.1686 show router rcf pending

revision 2

pendingChanges	*boolean*	There are pending routing control function changes.
[rcfCode]	*RcfPendingCodeUnitModel* see 2.1686.1	Pending routing control function code.
rcfCodeUnits	*table(string) = RcfPendingCodeUnitModel* see 2.1686.1	Pending routing control function code units keyed by unit name.

2.1686.1 RcfPendingCodeUnitModel

[rcfText]	*string*	Pending routing control function code unit text.
rcfTextModified	*boolean*	This pending code unit text has been modified.
[url]	*string*	Source file URL.

2.1687 show running-config [...]

see 2.938

2.1688 show running-config all [...]

see 2.938

2.1689 show running-config digest [...]

digest	*string*	Message digest for running-config.

2.1690 show schedule NAME | summary

jobs	*table(string) = ScheduleJobModels* see 2.1690.1	Maps job names to their respective details.
[maxJobs]	*integer*	Maximum number of jobs in progress at the same time.
[prependHostname]	*boolean*	Prepend host name to logfile.

2.1690.1 ScheduleJobModels

cliCommand	*string*	Scheduled Cli Command.
interval	*integer*	Cli command execution interval (in mins).
[jobInProgress]	*boolean*	Job in progress.

[jobInProgressStartTime]	*floating point number*	Cli commmand in progress execution start time (in secs).
[lastRun]	*floating point number*	Time the job last executed (in secs).
[lastRunStatus]	*integer*	Cli command last execution status.
logfileDirectory	*string*	Logfile directory location.
maxLogFiles	*integer*	Maximum number of logfiles stored.
maxTotalSize **	*integer*	Maximum total size of log files in bytes. **Since: 4.26.2
nextRun	*floating point number*	Next execution start time (in secs).
timeout	*floating point number*	Cli command execution timeout (in secs).
verbose	*boolean*	Verbose logging state.

2.1691 show security pki certificate rotation ROTATION_ID

[certificate]	*Certificate* see 2.948.1	Certificate imported into this rotation.
certificateSigningRequest	*CertificateSigningRequest* see 2.1691.1	The certificate signing request for the rotation.
creationTime	*integer*	Time rotation was created.
importTimeout	*integer*	Number of mins after rotation creation time that a certificate must be imported by.
profileName	*string*	SSL profile name for this rotation.
rotationId	*string*	Rotation ID.
rotationState	*enum: commitPending | importPending | commitExpired | importExpired*	Rotation states. importPending: Pending import of certificate. importExpired: Certificate was not imported within importTimeout window. commitPending: Certificate has been imported, waiting on commit. commitExpired: Certificate expired before rotation was comitted.

2.1691.1 CertificateSigningRequest

pemValue	*string*	CSR in PEM format.
publicKey	*PublicKey* see 2.948.1.3	Public key information.
subject	*DistinguishedName* see 2.948.1.1	Entity associated with the certificate.
version	*integer*	X.509 version.

2.1692 show security pki certificate rotation [...]

rotations	*table(string) = CertificateRotation* see 2.1691	A mapping between Rotation ID and rotation information.

2.1693 show segment-security [...]

revision 2

segments	*table(string) = Segment* see 2.1693.1.1	A mapping of Segment name to segment information.
vrfs	*table(string) = Vrf* see 2.1693.1	A mapping from VRF to segment security.

2.1693.1 Vrf

enabled	*boolean*	Indicates whether segment-security is enabled.
ipv6PrefixSupported **	*boolean*	IPv6 prefix supoort is enabled. **Since: 4.24.0
segments	*table(string) = Segment* see 2.1693.1.1	A mapping of segment name to segment information.
vrfExist	*boolean*	Whether VRF exist.

2.1693.1.1 Segment

classMap	*string*	Class Map of type segment-security associated with this segment.
fromSegments	*table(string) = string*	A mapping from source segment name to policy.
[interfaceVlans] **	*list of InterfaceVlanRange* see 2.1693.1.1.1	Interface plus VLAN ranges which are part of this segment. **Since: 4.23.1
interfaces	*list of Interface name*	Interfaces which are part of this segment.
ipv4PrefixListName	*string*	Name of IPv4 prefix list.
ipv6PrefixListName **	*string*	Name of IPv6 prefix list. **Since: 4.24.0

2.1693.1.1.1 InterfaceVlanRange

interface	*interface name*	Interface.
vlanRanges	*list of VlanRangeModel* see 2.1693.1.1.1.1	List of VLAN ranges associated with this interface for this segment.

2.1693.1.1.1.1 VlanRangeModel

endVlanId	*integer*	End of VLAN range.
startVlanId	*integer*	Start of VLAN range.

2.1694 show segment-security application [...]

revision 3

services	*table(string) = Service* see 2.1694.1	Segment Security applications.

2.1694.1 Service

dstPrefix	*list of IP address prefix*	Destination IP Prefix.
serviceDef	*list of ServiceDef* see 2.1694.1.1	Application definition.
srcPrefix	*list of IP address prefix*	Source IP Prefix.

2.1694.1.1 ServiceDef

allProtocols	*boolean*	Definitions supports all IP protocols.
dstPort	*list of PortRange* see 2.1694.1.1.1	destination port.
[ipProtocol]	*integer*	IP protocol number.
srcPort	*list of PortRange* see 2.1694.1.1.1	source port.

2.1694.1.1.1 PortRange

end	*integer*	Port range end.
start	*integer*	Port range start.

2.1695 show segment-security counters [...]

revision 2

[bypassPackets]	*integer*	Packets not destined to any segment.
dstSegments	*table(string) = DestSegment* see 2.1695.1	A mapping of destination segment to source segment.
[flowsCreated]	*integer*	Number of flows created.
[invalidPackets]	*integer*	Invalid packets dropped counter.
policies	*table(string) = PolicyCounter* see 2.1695.2	A mapping of policy names to counters for that policy.
[vrf]	*string*	VRF in which segment security is defined.
vrfExist	*boolean*	Whether VRF is created.

2.1695.1 DestSegment

srcSegments	*table(string) = SourceSegment* see 2.1695.1.1	A mapping of source segment to counters.

2.1695.1.1 SourceSegment

[defaultDrops] **	*integer*	No matching rule in policy. **Since: 4.29.2
[drops] **	*integer*	Explicit drop action counter. **Since: 4.29.2
hits	*integer*	Number of packets evaluated.

[policyName]	*string*	Policy name.

2.1695.2 PolicyCounter

defaultDrops	*integer*	No matching rule in policy.
drops	*integer*	Explicit drop action counter.
hits	*integer*	Number of packets evaluated.

2.1696 show segment-security errors [...]

vrfs	*table(string) = Vrf* see 2.1696.1	A mapping from VRF to its errors.

2.1696.1 Vrf

[dstSegments]	*table(string) = DestSegment* see 2.1696.1.1	Destination segments keyed by their name.
[error]	*string*	Error causing policies for this VRF to fail to install.

2.1696.1.1 DestSegment

srcSegments	*table(string) = SourceSegment* see 2.1696.1.1.1	A mapping of source segment name to its error.

2.1696.1.1.1 SourceSegment

error	*string*	Error causing this segment to fail to install.

2.1697 show segment-security hardware counters [...]

vrfs	*table(string) = VrfHardwareCounters* see 2.1697.1	A mapping of VRF name to its counters.

2.1697.1 VrfHardwareCounters

destinationSegments	*table(string) = DestinationSegmentCounters* see 2.1697.1.1	A mapping of destination segment to its counters.
policies	*table(string) = PolicyCounters* see 2.1697.1.2	A mapping of policy name to its counters.

2.1697.1.1 DestinationSegmentCounters

sourceSegments	*table(string) = SourceSegmentCounters* see 2.1697.1.1.1	A mapping of source segment to policy counters.

2.1697.1.1.1 SourceSegmentCounters

drops	*integer*	Policy drop counter.
hits	*integer*	Policy match counter.
policyName	*string*	Policy name.

2.1697.1.2 PolicyCounters

drops	*integer*	Drop action counter.
hits	*integer*	Forward action counter.

2.1698 show segment-security hardware detail [...]

vrfs	*table(string) = VrfHardwareDetail* see 2.1698.1	A mapping of VRF name to its segments.

2.1698.1 VrfHardwareDetail

segments	*table(string) = SegmentHardwareDetail* see 2.1698.1.1	A mapping of segment name to its hardware detail.

2.1698.1.1 SegmentHardwareDetail

hardwareId	*integer*	Hardware ID of security segment.
prefixes	*table(IPv4 or IPv6 address prefix) = SegmentPrefixHardwareDetail* see 2.1698.1.1.1	A mapping for segment prefix to its hardware information.

2.1698.1.1.1 SegmentPrefixHardwareDetail

l2AdjIndex	*integer*	Layer 2 adjacency index.

2.1699 show segment-security hardware routes [...]

vrfs	*table(string) = VrfHardwareRoutes* see 2.1699.1	A mapping of VRF name to its segments.

2.1699.1 VrfHardwareRoutes

segments	*table(string) = SegmentHardwareRoutes* see 2.1699.1.1	A mapping of segment name to its routes in hardware.

2.1699.1.1 SegmentHardwareRoutes

hardwareId	*integer*	Hardware ID of security segment.
prefixes	*table(IPv4 or IPv6 address prefix) = SegmentRouteDetail* see 2.1699.1.1.1	A mapping of route its hardware information.

2.1699.1.1.1 SegmentRouteDetail

routeType	*enum: S,F | S | F*	Route type.

2.1700 show segment-security hardware summary [...]

vrfs	*table(string) = VrfHardwareSummary* see 2.1700.1	A mapping of VRF name to its segments.

2.1700.1 VrfHardwareSummary

segments	*table(string) = SegmentHardwareSummary* see 2.1700.1.1	A mapping of segment name to its hardware summary.

2.1700.1.1 SegmentHardwareSummary

hardwareId	*integer*	Hardware ID of segment.
prefixesConfigured	*integer*	Number of prefixes configured.
prefixesProgrammed	*integer*	Number of prefixes programmed in hardware.

2.1701 show segment-security policy [...]

policies	*table(string) = Policy* see 2.1701.1	Segment Security Policy.

2.1701.1 Policy

policyDefs	*table(numeric string) = PolicyDef* see 2.1701.1.1	A mapping of sequence no to policy rule.
readonly **	*boolean*	Whether policy is readonly. **Since: 4.24.0

2.1701.1.1 PolicyDef

action	*enum: permit* | *statelessPermit* | *statelessDeny* | *deny*	Action to be applied on application.
log	*boolean*	Whether logging enabled.
serviceName	*string*	Application name.

2.1702 show segment-security status [...]

ipv6PrefixSupported	*boolean*	IPv6 prefix supoort is enabled.
vrfs	*table(string) = Vrf* see 2.1702.1	A mapping of VRF name to its segments status.

2.1702.1 Vrf

dstSegments	*table(string) = DestSegment* see 2.1702.1.1	Destination segments keyed by their name.

2.1702.1.1 DestSegment

ipv4PrefixListName	*string*	Name of IPv4 prefix list.
ipv6PrefixListName	*string*	Name of IPv6 prefix list.
srcSegments	*table(string) = SourceSegment* see 2.1702.1.1.1	A mapping of source segment name to its status.

2.1702.1.1.1 SourceSegment

status	*enum: success* | *waiting* | *failed*	Segment status.

2.1703 show service mss status

enabled	*boolean*	The Macro-Segmentation Service is enabled.
policyEnforcement	*MssPolicyEnforcementModel* see 2.1703.1	Policy enforcement properties.
running	*boolean*	The Macro-Segmentation Service is running.

2.1703.1 MssPolicyEnforcementModel

consistency	*enum: bestEffort* | *strict*	Policy enforcement mode.
rules	*MssPolicyEnforcementRulesModel* see 2.1703.1.1	Policy enforcement rules.

2.1703.1.1 MssPolicyEnforcementRulesModel

group	*boolean*	Group policy enforcement rule enabled.
verbatim	*boolean*	Verbtim policy enforcement rule enabled.

2.1704 show service pcs controller group [...]

nsGroups	*list of ControllerNsGroup* see 2.1704.1	List of NSX security groups received from the controller.

2.1704.1 ControllerNsGroup

addrList	*list of string*	List of IP/IPv6 address making up the NS group.
name	*string*	NS group name.

2.1705 show service pcs controller host IP_GEN_ADDR

groups	*list of string*	List of NSX security groups the host belongs to.
tags	*list of string*	List of tags assigned to the host.

2.1706 show service pcs controller policy [...]

policies	*list of ControllerPolicy* see 2.1706.1	List of policies received from controller.

2.1706.1 ControllerPolicy

priority	*integer*	Priority of the section.
rules	*list of ControllerRule* see 2.1706.1.1	List of rules of the policy section.
sectionId	*string*	Section ID.
sectionName	*string*	Name of the policy section.

2.1706.1.1 ControllerRule

action	*enum: drop* | *allow* | *unknown* | *reject*	Action of the rule.
destinationPort	*string*	Destination port(s).
destinations	*list of string*	List of destinations.
priority	*integer*	Priority of the rule within a section.
protocol	*enum: unknown* | *udp* | *icmp* | *tcp* | *ip*	Protocol.
ruleId	*string*	Rule ID.
ruleName	*string*	Name.
sourcePort	*string*	Source port(s).

sources	*list of string*	List of sources.

2.1707 show service pcs policy [...]

section	*list of PolicySectionStatus* see 2.1707.1	A list of policy status.

2.1707.1 PolicySectionStatus

sectionId	*string*	Section ID.
sectionName	*string*	Name of the policy section.
status	*enum: pending | success | unknown | fail*	Status of the policy.
switch	*list of PolicySectionSwitchStatus* see 2.1707.1.1	Section status on switch.

2.1707.1.1 PolicySectionSwitchStatus

acl	*list of SwitchAclStatus* see 2.1707.1.1.1	ACLs of the switch covering the policy.
switchId	*string*	Switch name or ID.

2.1707.1.1.1 SwitchAclStatus

aclName	*string*	Name of the ACL that covers the policy.
status	*enum: pending | success | unknown | fail*	Status of the ACL on the switch.

2.1708 show service pcs policy counters [...]

policies	*list of PolicyCounters* see 2.1708.1	Status and counter information of policies.

2.1708.1 PolicyCounters

policyId	*string*	Policy ID.
policyName	*string*	Name of the policy.
rules	*list of RuleCounters* see 2.1708.1.1	Status and counter information of policy rules.
status	*enum: pending | success | unknown | fail*	Status of the policy.

2.1708.1.1 RuleCounters

packetCount	*integer*	Number of packets matched the rule across the switches.
ruleId	*string*	Rule ID.
ruleName	*string*	Name of the rule.

switchError	*list of PolicySwitchErrorMsg* see 2.1708.1.1.1	Errors occurred on the switches.

2.1708.1.1.1 PolicySwitchErrorMsg

error	*string*	Error occurred when programming policy on the switch.
switchId	*string*	Switch name or ID.

2.1709 show service pcs status

[apiState]	*table(string) = PcsApiState* see 2.1709.1	A dictionary of API states, keyed by API type.
policyCount	*integer*	Total number of policies received from controller.
status	*enum: running | notRunning*	Status of Policy Control Service.
tagCount	*integer*	Total number of tags configured on switches.

2.1709.1 PcsApiState

[errorDetail]	*string*	Additional detail for error.
method	*enum: post | delete | get | put*	HTTP method.
success	*boolean*	Successful HTTP request or response.
timestamp	*floating point number*	UTC timestamp of the last response.
uri	*string*	HTTP URI.

2.1710 show service pcs tag [...]

tags	*table(string) = TagInfo* see 2.1710.1	List of tags configured on the switches.

2.1710.1 TagInfo

switches	*table(string) = SwitchInfo* see 2.1710.1.1	List of switches the tag is configured on.
tag	*string*	Tag.

2.1710.1.1 SwitchInfo

intfs	*table(string) = IntfInfo* see 2.1710.1.1.1	List of interfaces the host is attached to.
switchId	*string*	Switch ID or hostname.

2.1710.1.1.1 IntfInfo

hosts	*list of string*	List of hosts learnt on the interface.
name	*interface name*	Interface name.

2.1711 show service vxlan address-table advertised [...]

revision 2

vniInDottedNotation	*boolean*	VNI display format.
vnis	*list of VxlanVniStatus* see 2.1711.1	list of Vxlan Vni Status, one per VNI.

2.1711.1 VxlanVniStatus

bumVtepListTable	*list of VxlanMacVtepListPair* see 2.1711.1.1	Flood table.
macAddress	*MAC address*	A MAC address.
unicastHostTable	*list of VxlanMacVtepPair* see 2.1711.1.2	Unicast address table.
[vlan]	*integer*	VLAN.
[vni]	*integer*	VNI.

2.1711.1.1 VxlanMacVtepListPair

macAddr	*MAC address*	Destination MAC address.
vtepIpList	*list of IPv4 address*	List of VTEPs.
vtepIpListType	*string*	BUM forwarding mode.

2.1711.1.2 VxlanMacVtepPair

macAddr	*MAC address*	Host Mac address.
[moveCount]	*integer*	Number of local and remote Mac moves.
vtepIp	*IPv4 address*	Vtep Ip address.

2.1712 show service vxlan address-table received [...]

revision 2

nonSwitchSources **	*table(string) = VxlanVniStatusList* see 2.1711	Mapping from non-switch source to VniStatus. **Since: 4.21.3
switches	*table(string) = VxlanVniStatusList* see 2.1711	Mapping from switch id to VniStatus.

2.1713 show service vxlan arp advertised [...]

`vnis`	*table(numeric string) = VxlanArpTable* see 2.1713.1	A mapping from VNI to ARP table.

2.1713.1 VxlanArpTable

`addresses`	*table(IPv4 or IPv6 address) = VxlanArpEntry* see 2.1713.1.1	A mapping from IP address to ARP entry.

2.1713.1.1 VxlanArpEntry

`changes`	*integer*	Number of times the ARP entry changed.
`macAddress`	*MAC address*	MAC address of an ARP entry.
`preference`	*integer*	ARP entry preference value.

2.1714 show service vxlan arp received [...]

`switches`	*table(string) = VxlanVtep* see 2.1714.1	A mapping from switch ID to vtep.

2.1714.1 VxlanVtep

`vnis`	*table(string) = VxlanArpTable* see 2.1713.1	A mapping from VNI to ARP table.

2.1715 show service vxlan nd advertised [...]

see 2.1713

2.1716 show service vxlan nd received [...]

see 2.1714

2.1717 show service vxlan status

`arpRelayVteps` **	*list of string*	List of VTEPs relaying ARP replies. **Since: 4.18.0
`[bgpEvpnRedistEnabled]` **	*boolean*	When set to True it means MAC-VTEP bindings generated by BGP EVPN are being redistributed in to Vxlan Control Service. **Since: 4.21.3
`[capability]` **	*VxlanControllerCapability* see 2.1717.1	Vxlan Controller capability. **Since: 4.15.3

macLearning	*enum: dataPlane | controlPlane*	MAC learning. Defaults to controlPlane.
[multicastDistribution-Enabled] **	*boolean*	When set to True it signifies multicast floodlist distribution happens on Vxlan Controller. **Since: 4.21.3
[resyncInProgress]	*boolean*	When set to True it signifies Vxlan Controller Service sync, after CVX restart, is in progress.
resyncPeriod	*integer*	Sync period in seconds for Vxlan Controller Service after CVX restart. Value of 0 means finish sync immediately after CVX reboot without waiting.
[resyncStartedAt]	*floating point number*	UTC Time at which sync of Vxlan Controller Service started after CVX restart. 0 signifies never.
status	*enum: running | shutting down | stopped*	Status of VxlanController agent.
vniMapping **	*enum: vlan | portDot1qTag*	VNI mapping. Defaults to vlan. **Since: 4.20.5

2.1717.1 VxlanControllerCapability

indirectRouting **	*boolean*	Indicates indirect routing support. **Since: 4.18.0
routing	*boolean*	Indicates routing support.

2.1718 show service vxlan vni advertised [...]

switchList	*table(string) = VniList* see 2.1718.1	List of VNIs advertised per switch.
vniDotted	*boolean*	VNI display format.

2.1718.1 VniList

converging	*boolean*	True if Vxlan Controller Service sync in progress.
vnis	*list of VniRange* see 2.1718.1.1	List of VNIs in dotted or plain notation.

2.1718.1.1 VniRange

maxVni	*integer*	Maximum value for VNI range.
minVni	*integer*	Minimum value for VNI range.

2.1719 show service vxlan vni received [...]

switchList	*table(string) = VniList* see 2.1718.1	List of switches and VNIs received from each.
vniDotted	*boolean*	VNI display format.

vniList	*table(string) = SwitchList* see 2.1719.1	List of VNIs and switches they were received from.

2.1719.1 SwitchList

switches	*list of string*	List of Switches, hostname, IP or default ID.

2.1720 show service-insertion connection [...]

connectionStatus	*list of SfeSiConnStatusModel* see 2.1720.1	All Connection Status.

2.1720.1 SfeSiConnStatusModel

connName	*string*	Connection name.
[connectivityMonitor]	*string*	Configured connectivity monitor hostname.
isActive	*boolean*	Connection is active or not.
[nextHop]	*string*	Nexthop of this exit.
[nhResolved]	*boolean*	Nexthop is resolved.
[primaryIntf]	*SfeSiConnIntfStatusModel* see 2.1720.1.1	Configured primary interface.
[secondaryIntf]	*SfeSiConnIntfStatusModel* see 2.1720.1.1	Configured secondary interface.

2.1720.1.1 SfeSiConnIntfStatusModel

intfEnabled	*boolean*	Interface is enabled.
intfId	*string*	Configured interface.
[monHostReachable]	*boolean*	Monitor host is reachable.
vrf	*string*	Configured VRF on this interface.

2.1721 show session-config [...]

see 2.938

2.1722 show sflow [...]

[accelEnabled] **	*boolean*	sFlow hardware acceleration running. **Since: 4.21.3
[accelSampleRate] **	*integer*	Sample rate for hardware acceleration, 1 sample per N packets. **Since: 4.21.3
[accelSupported] **	*boolean*	Is hardware acceleration supported. **Since: 4.21.3
[bgpExports]	*list of BgpExport* see 2.1722.1	List of BGP Exports.
datagrams	*integer*	Number of datagrams sent for regular sFlow samples.
[details]	*Details* see 2.1722.7	Detailed sFlow information.
[dropPacketExportRunning] **	*boolean*	sFlow export is running for drop samples. **Since: 4.30.1
[dropPacketSamplingEnabled] **	*boolean*	sFlow export enabled for dropped packets. **Since: 4.30.1
[dropSampleDatagrams] **	*integer*	Number of datagrams sent for drop samples. **Since: 4.30.1
[dropSamplesReceived] **	*integer*	Number of drop samples received at sFlow Agent. **Since: 4.30.1
[dropSamplesSent] **	*integer*	Number of drop samples sent to external collectors from sFlow Agent. **Since: 4.30.1
dscpValue **	*integer*	DSCP value in sFlow datagram. **Since: 4.24.1
enabled	*boolean*	sFlow globally enabled.
[hardwareAccelSampleRate] **	*integer*	Hardware sample rate for hardware accelerated Sflow, 1 sample per N packets. **Since: 4.21.3
hardwareSampleRate	*integer*	Hardware sample rate for software Sflow, 1 sample per N packets.
hardwareSamples	*integer*	Number of hardware samples.
[hwSampleTruncateSize] **	*integer*	HW Sample truncate size limit in bytes. **Since: 4.24.2
ipv4Destinations	*list of Ipv4Destination* see 2.1722.2	List of IPv4 destinations.
[ipv4DestinationsForDropSamples] **	*list of Ipv4Destination* see 2.1722.2	List of IPv4 destinations for drop samples. **Since: 4.30.1
ipv4Sources	*list of Ipv4Source* see 2.1722.3	List of Source IPv4 addresses.
[ipv4SourcesForDropSamples] **	*list of Ipv4Source* see 2.1722.3	List of Source IPv4 addresses for drop samples. **Since: 4.30.1
ipv6Destinations	*list of Ipv6Destination* see 2.1722.4	List of IPv6 destinations.
[ipv6DestinationsForDropSamples] **	*list of Ipv6Destination* see 2.1722.4	List of IPv6 destinations for drop samples. **Since: 4.30.1

ipv6Sources	*list of Ipv6Source* see 2.1722.5	List of Source IPv6 addresses.
[ipv6SourcesForDropSamples] **	*list of Ipv6Source* see 2.1722.5	List of Source IPv6 addresses for drop samples. **Since: 4.30.1
polling	*boolean*	sFlow polling.
pollingInterval	*floating point number*	Polling interval in seconds.
[rewriteDscp]	*boolean*	Rewrite DSCP value in sFlow samples.
running **	*boolean*	sFlow sampling is running. **Since: 4.30.1
samplePool	*integer*	Sample pool.
sampleRate	*integer*	Sample rate, 1 sample per N packets.
sampleTruncateSize **	*integer*	Sample truncate size limit in bytes. **Since: 4.23.2
samplingEnabled	*boolean*	sFlow sampling enabled.
sendingDatagrams	*list of SendingDatagram* see 2.1722.6	List of Sending datagrams.
[sendingDatagramsForDrop-Samples] **	*list of SendingDatagram* see 2.1722.6	List of sending datagrams for drop samples. **Since: 4.30.1
softwareSamples	*integer*	Number of samples.
totalPackets	*integer*	Total packets.

2.1722.1 BgpExport

export	*boolean*	Export BGP information.
vrfName	*string*	VRF name.

2.1722.2 Ipv4Destination

datagramsCount **	*integer*	Number of datagrams sent to sFlow collector. **Since: 4.30.1
hostname	*string*	Hostname of sFlow collector.
ipv4Address	*IPv4 address*	IPv4 address of sFlow collector.
port	*integer*	Port of sFlow collector.
vrfName	*string*	Associated VRF for sFlow collector.

2.1722.3 Ipv4Source

ipv4Address	*IPv4 address*	Source IPv4 address.
[sourceInterface]	*interface name*	Source Interface.
vrfName	*string*	Associated VRF for sFlow source.

2.1722.4 Ipv6Destination

datagramsCount **	*integer*	Number of datagrams sent to sFlow collector. **Since: 4.30.1
hostname	*string*	Hostname of sFlow collector.
ipv6Address	*IPv6 address*	IPv6 address of sFlow collector.
port	*integer*	Port of sFlow collector.
vrfName	*string*	Associated VRF for sFlow collector.

2.1722.5 Ipv6Source

ipv6Address	*IPv6 address*	Source IPv6 address.
[sourceInterface]	*interface name*	Source Interface.
vrfName	*string*	Associated VRF for sFlow source.

2.1722.6 SendingDatagram

reason	*enum: bindingInProgress* | *notRunning* | *noError* | *invalidVrf* | *invalidDestination* | *invalidSourceDestination* | *invalidSource*	Reason for sendingDatagrams state.
sending	*boolean*	sFlow samples being sent.
vrfName	*string*	VRF name.

2.1722.7 Details

[accelUnsupported-Extensions] **	*list of string*	Extensions that are not supported for hardware acceleration. **Since: 4.21.3
hardwareSamplingEnabled	*boolean*	Hardware sampling enabled.
[portChannelOutputIfIndex] **	*enum: member* | *portchannel*	Output interface index for port channels. **Since: 4.22.0
[sampleEgressSubintf] **	*boolean*	Include output subinterface information in packet sample; sample sent in expanded format. **Since: 4.22.1
[sampleEncodingFormat] **	*enum: compact* | *expanded*	Encoding format for samples. **Since: 4.22.1
[sampleIngressSubintf] **	*boolean*	Include input subinterface information in packet sample; sample sent in expanded format. **Since: 4.23.0
[sampleMplsExtension] **	*boolean*	Include MPLS extension header in packet sample. **Since: 4.22.0
sampleOutputInterface	*boolean*	Include output interface in packet sample.
[sampleRouterExtension]	*boolean*	Include router extension header in packet sample.

[sampleSwitchExtension]	*boolean*	Include switch extension header in packet sample.
[sampleTunnelIpv4Egr-Extension] **	*boolean*	Include tunnel IPv4 egress extension header in packet sample. **Since: 4.22.0
[sampleVxlanHeaderStrip] **	*boolean*	Include VXLAN tunnel headers of VXLAN encapsulated packets in packet sample. **Since: 4.28.0
samplesDiscarded	*integer*	Number of discarded samples.

2.1723 show sflow hardware accelerators

accelerators	*table(string) = SflowAcceleratorInfo* see 2.1723.1	Mapping of accelerator name to accelerator info.

2.1723.1 SflowAcceleratorInfo

accelRevision **	*integer*	Accelerator revision. **Since: 4.22.0
chips	*list of string*	List of chips directly connected to this accelerator.
pciAddress	*string*	PCI address of this accelerator.
sflowAccelFpgaType	*string*	Type of sFlow accelerator.
sliceName	*string*	Name of the slice on which this sFlow accelerator is located.

2.1724 show sflow hardware counters

counters	*table(string) = HardwareSflowCountersInfo* see 2.1724.1	sFlow accelerator counters.

2.1724.1 HardwareSflowCountersInfo

inPktCnt	*integer*	Incoming Packet Count.
inPktErrCnt	*integer*	Incoming Packet Error Count.
inProcPktCnt	*integer*	Incoming Processed Packet Count.
outFlowSampleCnt	*integer*	Outgoing Flow Sample Count.
outProcSflowDgramCnt	*integer*	Outgoing Processed Datagram Count.
outSflowDgramCnt	*integer*	Outgoing Sflow Datagram Count.
rcvPktDropCnt	*integer*	Receive Packet Drop Count.
rcvPktTrunCnt	*integer*	Packet Truncated Count.
samplePool	*integer*	Sample Pool.

2.1725 show sflow hardware mapping

chips	*table(string) = SflowAcceleratorMappingInfo* see 2.1725.1	Mapping of chip to sFlow accelerator.

2.1725.1 SflowAcceleratorMappingInfo

directConnection	*boolean*	Direct connection between the chip and the accelerator.
sflowAccelerator	*string*	sFlow accelerator name.

2.1726 show sflow hardware mapping lag

lags	*table(string) = SflowAcceleratorMappingLagInfo* see 2.1726.1	Mapping of LAG to sFlow accelerator.

2.1726.1 SflowAcceleratorMappingLagInfo

sflowAccelerator	*string*	sFlow accelerator name.

2.1727 show sflow hardware status

accelPPSupported **	*boolean*	Whether sFlow hardware acceleration in pipeline mode is available on the system. **Since: 4.30.0
[inactiveFeatures]	*list of string*	List of sFlow features that are configured but not supported with hardware-accelerated sFlow, and will only take effect with software sFlow.
[modules]	*table(string) = SflowAccelModuleStatus* see 2.1727.2	Per-module status of hardware-accelerated sFlow.
[reasonsNotRunning]	*list of string*	List of reasons why sFlow hardware acceleration is not running.
running	*boolean*	Whether sFlow hardware acceleration is running.
sampleRate	*integer*	Sample rate, 1 sample per N packets.
vrfs **	*table(string) = Vrf* see 2.1727.1	A mapping from VRF's name to its collectors. **Since: 4.25.2

2.1727.1 Vrf

collectorIps	*table(IPv4 or IPv6 address) = Collector* see 2.1727.1.1	A mapping of collector IP address to its status.

2.1727.1.1 Collector

active	*boolean*	Whether this collector is active.
port	*integer*	Port of sFlow collector.
reason	*enum: noHardwareResource | noRouteToCollector | ip6CollectorInUse | enabled | missingConfig*	Reason why collector is disabled it's an empty string if collector is enabled.

2.1727.2 SflowAccelModuleStatus

hasAccelerators	*boolean*	Whether this module has sFlow accelerators.
sflowMode	*ModuleSflowMode* see 2.1727.2.1	Type of sFlow running for this module.

2.1727.2.1 ModuleSflowMode

active	*enum: disabled | hardware-accelerated | software*	Current sFlow status of this module.
configured	*enum: software | hardware-accelerated*	Whether this module is configured to use software or hardware-accelerated sFlow.

2.1728 show sflow interfaces [...]

revision 3

egressInterfacesEnabled-Default	*boolean*	Global egress sFlow configuration for interfaces.
enabled	*boolean*	sFlow globally enabled.
ingressInterfacesEnabled-Default	*boolean*	Global ingress sFlow configuration for interfaces.
interfaces	*table(Interface name) = SflowTypeStatusMap* see 2.1728.1.1	Mapping between an interface and its status.
lagInterfaces	*table(Interface name) = LagMemberSflowTypeStatus* see 2.1728.1	Mapping between a LAG member and its status.

2.1728.1 LagMemberSflowTypeStatus

lagInterface	*interface name*	LAG interface for LAG member.
lagSflowTypeStatus	*SflowTypeStatusMap* see 2.1728.1.1	sFlow type status map for LAG member.

2.1728.1.1 SflowTypeStatusMap

[intfOperStatusReason]	*string*	Operational status reason.
sFlows	*table(string) = SflowTypeStatus* see 2.1728.1.1.1	A mapping of sFlow type to its status.

2.1728.1.1.1 SflowTypeStatus

status	*enum: running | partial | inactive*	sFlow status for an interface.

2.1729 show sflow interfaces counters

enabled	*boolean*	sFlow globally enabled.
interfaces	*table(Interface name) = IntfSampleCounterMap* see 2.1729.1	Mapping between an interface and its sample counters.

2.1729.1 IntfSampleCounterMap

[egress]	*integer*	Interface egress sample count.
[ingress]	*integer*	Interface ingress sample count.

2.1730 show snmp

chassis	*SnmpChassis* see 2.1730.2	Chassis ID.
contact	*Contact* see 2.1730.1	SNMP contact.
[counters]	*SnmpCounters* see 2.1730.3	SNMP counters.
enabled	*boolean*	Indicates whether SNMP is enabled.
location	*SnmpLocation* see 2.1730.5	SNMP location.
logging	*SnmpLogging* see 2.1730.6	SNMP logging.
misc	*SnmpMiscConfig* see 2.1730.7	SNMP miscellaneous configuration.
srcIntf	*SnmpInterface* see 2.1730.4	SNMP source interfaces.
vrfs	*SnmpVrfs* see 2.1730.8	SNMP VRFs.

2.1730.1 Contact

contact	*string*	SNMP Contact.

2.1730.2 SnmpChassis

chassisId	*string*	Chassis ID.

2.1730.3 SnmpCounters

groupLength	*integer*	Users in Group.
inBadCommunityNames	*integer*	Unknown community name.
inBadCommunityUses	*integer*	Illegal operation for community name.
inGetNextPdus	*integer*	Get-next PDUs.
inGetPdus	*integer*	Get-request PDUs.
inParseErrs	*integer*	Encoding errors.
inPkts	*integer*	SNMP packet input.
inRequestVars	*integer*	Requested variables.
inSetPdus	*integer*	Set-request PDUs.
inSetVars	*integer*	Altered variables.
inVersionErrs	*integer*	Bad SNMP version errors.
numberOfGroups	*integer*	Groups.
numberOfViews	*integer*	Views.
outBadValueErrs	*integer*	Bad value errors.
outGeneralErrs	*integer*	General errors.
outGetResponsePdus	*integer*	Response PDUs.
outNoSuchNameErrs	*integer*	No such name errors.
outPkts	*integer*	SNMP packets output.
outTooBigErrs	*integer*	Too big errors.
outTrapPdus	*integer*	Trap PDUs.

serviceEnabled	*boolean*	status Snmp agent and snmpd.

2.1730.4 SnmpInterface

sourceInterfaces **	*table(string) = string*	A mapping between a VRF and the source interface. **Since: 4.17.0

2.1730.5 SnmpLocation

location	*string*	SNMP Location.

2.1730.6 SnmpLogging

hosts	*table(string) = SnmpLoggingHost* see 2.1730.6.1	Mapping between hostname and host information.
loggingEnabled	*boolean*	If logging is enabled.

2.1730.6.1 SnmpLoggingHost

port	*integer*	Port for SNMP logger.
[vrf]	*string*	VRF, if applicable, for SNMP logger.

2.1730.7 SnmpMiscConfig

transmitMsgMaxSize	*integer*	Maximum number of bytes in SNMP message (UDP/TCP payload).

2.1730.8 SnmpVrfs

snmpVrfs	*list of string*	List of VRFs the SNMP agent is enabled on.

2.1731 show snmp ((ipv4 access-list) | (ipv6 access-list)) [...]

see 2.53

2.1732 show snmp (notification | trap)

notifications	*list of SnmpNotification* see 2.1732.1	List of notifications that are configured.

2.1732.1 SnmpNotification

component	*string*	Component owning notification.
enabled	*boolean*	Notification is enabled.
name	*string*	Name of notification.
reason	*string*	Reason for notification state.

2.1733 show snmp chassis

see 2.1730.2

2.1734 show snmp community

communities	*table(string) = SnmpCommunityDetailModel* see 2.1734.1	A mapping of SNMP community to detail community information.

2.1734.1 SnmpCommunityDetailModel

access	*enum: readWrite | readOnly*	Community access level.
defaultContext **	*string*	Context associated for this community. **Since: 4.29.1
ipV4AccessList	*string*	IPv4 access-list for this community.
[ipV4AccessListStatus]	*enum: ignoredNotAStandardAcl | nonExistent*	IPv4 access list status.
ipV6AccessList	*string*	IPv6 access-list for this community.
[ipV6AccessListStatus]	*enum: ignoredNotAStandardAcl | nonExistent*	IPv6 access list status.
view	*string*	Community MIB view name.
[viewStatus]	*notConfigured*	Snmp view status.

2.1735 show snmp contact

see 2.1730.1

2.1736 show snmp debug counters table (TABLE | all) [...]

contexts	*table(string) = SnmpTableIteratorCounterModel* see 2.1736.1	Counters by context.

2.1736.1 SnmpTableIteratorCounterModel

tables	*table(string) = SnmpTableIteratorCounterType* see 2.1736.1.1	Counters by table.

2.1736.1.1 SnmpTableIteratorCounterType

counterTypes	*table(string) = SnmpTableIteratorCounter* see 2.1736.1.1.1	Table iterator counters by type.

2.1736.1.1.1 SnmpTableIteratorCounter

counters	*table(string) = integer*	Table iterator counters.

2.1737 show snmp debug identifier hashed

hashedIdentifiers	*table(string) = string*	A mapping of actual identifier to hashed identifier.

2.1738 show snmp engineID

localEngineId	*string*	Local SNMP EngineID.
remoteEngineIds	*SnmpRemoteEngineIdHostToPortModel* see 2.1738.1	Remote SNMP EngineIDs.

2.1738.1 SnmpRemoteEngineIdHostToPortModel

hosts	*table(string) = SnmpRemoteEngineIdPortToEngIdModel* see 2.1738.1.1	A mapping of hostname to a collection of UDP-ports.

2.1738.1.1 SnmpRemoteEngineIdPortToEngIdModel

ports	*table(numeric string) = string*	A mapping of UDP-port to engineID.

2.1739 show snmp group [...]

groups	*table(string) = SnmpGroupVerToDetailModel* see 2.1739.1	A mapping of SNMP group name to SNMP versions.

2.1739.1 SnmpGroupVerToDetailModel

versions	*table(string) = SnmpGroupDetailModel* see 2.1739.1.1	A mapping of SNMP version to SNMP group details.

2.1739.1.1 SnmpGroupDetailModel

notifyView	*string*	Name of the SNMP notify view.
[notifyViewConfig]	*boolean*	True if the notify view exists and has been configured.
readView	*string*	Name of the SNMP read view.
[readViewConfig]	*boolean*	True if the read view exists and has been configured.
secModel	*enum: v3Priv | v2c | v1 | v3Auth | v3NoAuth*	Security model in effect for this group.
writeView	*string*	Name of the SNMP write view.
[writeViewConfig]	*boolean*	True if the write view exists and has been configured.

2.1740 show snmp host

hosts	*list of HostModel* see 2.1740.1	SNMP notification recipients.

2.1740.1 HostModel

hostname	*string*	Host name or IP address.
notificationType	*enum: inform | trap*	SNMP notification type.
port	*integer*	UDP port number.
protocolVersion	*enum: v2c | v1 | v3*	SNMP version.
[v1v2cParams]	*SnmpV1V2cHostParams* see 2.1740.1.1	SNMPv1/v2c parameters.
[v3Params]	*SnmpV3HostParams* see 2.1740.1.2	SNMPv3 parameters.
[vrf]	*string*	Name of the VRF.

2.1740.1.1 SnmpV1V2cHostParams

communityString	*string*	Community string for SNMPv1/v2c.

2.1740.1.2 SnmpV3HostParams

securityLevel	*enum: authPriv | authNoPriv | noAuthNoPriv*	SNMPv3 security level.
user	*string*	SNMPv3 user.

2.1741 show snmp local-interface

see 2.1730.4

2.1742 show snmp location

see 2.1730.5

2.1743 show snmp mib

contexts	*table(string) = SnmpOidsModel* see 2.1743.1	A mapping from a context (VRF) to its registered OIDs.

2.1743.1 SnmpOidsModel

oids	*list of string*	Registered OIDs.

2.1744 show snmp mib bridgemib portid [...]

portIDs	*table(string) = integer*	Map an interface name to a port id.

2.1745 show snmp mib ifmib ifindex [...]

ifIndex	*table(Interface name) = integer*	Mapping between interface name and the corresponding SNMP interface index.
pending **	*list of Interface name*	Interfaces for which an ifIndex value has not yet been allocated. **Since: 4.24.0
reservedIfIndex **	*table(Interface name) = integer*	Mapping from interface name to the corresponding SNMP interface index reserved for use by an interface that has been deleted, if it is recreated in the future. **Since: 4.28.2

2.1746 show snmp mib qosmib counter-interval

enabled	*boolean*	QOS-MIB counters are enabled.
interval	*integer*	QOS-MIB counter update interval (seconds).

2.1747 show snmp mib translate OID

numeric	*string*	Numeric form of the OID.
symbolic	*string*	Symbolic (textual) form of the OID.

2.1748 show snmp net-snmp buffer

clientRecvBuf	*integer*	Client receive buffer size.
clientSendBuf	*integer*	Client send buffer size.
serverRecvBuf	*integer*	Server receive buffer size.
serverSendBuf	*integer*	Server send buffer size.

2.1749 show snmp notification host

see 2.1740

2.1750 show snmp scheduler [...]

see 2.79

2.1751 show snmp source-interface

see 2.1730.4

2.1752 show snmp user [...]

usersByVersion	*table(string) = UserCollectionModel* see 2.1752.1	SNMP users grouped by version v1, v2c or v3.

2.1752.1 UserCollectionModel

users	*table(string) = UserModel* see 2.1752.1.1	SNMP users.

2.1752.1.1 UserModel

groupConfigured	*boolean*	Indicates whether SNMP group is configured.
groupName	*string*	Group for the SNMP user.
[v3Params]	*SnmpV3UserParams* see 2.1752.1.1.1	SNMPv3 parameters for the user.

2.1752.1.1.1 SnmpV3UserParams

[authType]	*enum: SHA-384 | MD5 | SHA | SHA-512 | SHA-256 | SHA-224*	Authentication protocol for the user.
engineId	*string*	SNMPv3 engineID.
engineIdMatch **	*boolean*	The system engineID matches the user. **Since: 4.27.1

[privType]	*enum: AES-128 | AES-256 | DES | AES-192*	Privacy protocol for the user.

2.1753 show snmp v2-mib chassis

see 2.1730.2

2.1754 show snmp v2-mib contact

see 2.1730.1

2.1755 show snmp v2-mib location

see 2.1730.5

2.1756 show snmp view [...]

views	*list of SnmpView* see 2.1756.1	List of SNMP views.

2.1756.1 SnmpView

name	*string*	SNMP view name.
viewType	*table(string) = boolean*	Mapping from OID subtree to inclusion status.

2.1757 show spanning-tree [...]

spanningTreeInstances	*table(string) = SpanningTreeInstance* see 2.1757.1	Map STP instance names to their details.

2.1757.1 SpanningTreeInstance

bridge	*SpanningTreeBridge* see 2.1757.1.1	Spanning tree bridge details.
[detail]	*SpanningTreeInstance-Detail* see 2.1757.1.2	Spanning tree instance details.
[instanceName]	*string*	Name of spanning tree instance.
interfaces	*table(Interface name) = SpanningTreeInterface* see 2.1757.1.3	Map STP interface names to their details.
protocol	*enum: rapidPvst | rstp | mstp*	Spanning tree protocol configured.
[rootBridge]	*SpanningTreeBridge* see 2.1757.1.1	Spanning tree root bridge details.
[rootPort]	*SpanningTreeRootPort* see 2.1757.1.4	Root Port.

2.1757.1.1 SpanningTreeBridge

[detail]	*SpanningTreeBridgeDetail* see 2.1757.1.1.1	Spanning tree bridge details.
forwardDelay	*integer*	Forward delay for the bridge (seconds).
helloTime	*floating point number*	Hello time for the bridge (seconds).
macAddress	*MAC address*	MAC address of the bridge.
maxAge	*integer*	Max age for the bridge (seconds).
priority	*integer*	Priority of the bridge (0 to 61440 in increments of 4096).
systemIdExtension	*integer*	System Id extension.

2.1757.1.1.1 SpanningTreeBridgeDetail

transmitHoldCount	*integer*	Transmit Hold Count. This is the burstiness for transmitted BPDUs. TransmitHoldCount number of BPDUs are allowed to be transmitted every (2000/(Hello Time * TransmitHoldCount)) seconds, where Hello Time is in milliseconds.

2.1757.1.2 SpanningTreeInstanceDetail

[lastTopologyChangePort]	*interface name*	Port on which last topology change happened.
[lastTopologyChangeTime]	*integer*	Seconds since last topology change.
topologyChanges	*integer*	Number of topology changes occurred.

2.1757.1.3 SpanningTreeInterface

boundaryType	*enum: mstpRegionPvst | mstpRegion | stp | none*	Spanning tree boundary type.
cost	*integer*	Spanning tree path cost.
[detail]	*SpanningTreeInterface-Detail* see 2.1757.1.3.2	Spanning tree interface details.
inconsistentFeatures	*SpanningTreeInconsistent-Features* see 2.1757.1.3.1	Spanning tree inconsistent features.
[isEdgePort]	*boolean*	Interface is an edge port.
linkType	*enum: shared | p2p*	Spanning tree link type.
[mappedVlans]	*VlanList* see 2.1757.1.3.5	Vlans mapped to the MST instance interface.
portNumber	*integer*	Port number.
priority	*integer*	Spanning tree port priority.
role	*enum: backup | master | designated | disabled | root | alternate*	Spanning tree port role.
state	*enum: learning | forwarding | discarding*	Spanning tree port state.

2.1757.1.3.1 SpanningTreeInconsistentFeatures

bridgeAssurance	*boolean*	Bridge assurance is inconsistent.
loopGuard	*boolean*	Loop Guard is inconsistent.
[mstPvstBorder]	*boolean*	MST-PVST border is inconsistent.
rootGuard	*boolean*	Root Guard is inconsistent.

2.1757.1.3.2 SpanningTreeInterfaceDetail

Could also be an instance of:
SpanningTreeInterfaceAllDetail 2.1757.1.3.3
SpanningTreeInterfaceMstDetail 2.1757.1.3.4

designatedBridgeAddress	*MAC address*	Address of designated bridge.
designatedBridgePriority	*integer*	Priority of designated bridge.
designatedPathCost	*SpanningTreePathCost* see 2.1757.1.3.2.1	Designated path cost.
designatedPortNumber	*integer*	Port number of designated port.
designatedPortPriority	*integer*	Priority of designated port.
designatedRootAddress	*MAC address*	Address of designated root bridge.
designatedRootPriority	*integer*	Priority of designated root bridge.

2.1757.1.3.2.1 SpanningTreePathCost

Could also be an instance of:
SpanningTreePathCostCist 2.1757.1.3.2.2
SpanningTreePathCostDefault 2.1757.1.3.2.3

2.1757.1.3.2.2 SpanningTreePathCostCist

externalCost	*integer*	External Root Path Cost.
internalCost	*integer*	Internal Root Path Cost.

2.1757.1.3.2.3 SpanningTreePathCostDefault

cost	*integer*	Root Path Cost.

2.1757.1.3.3 SpanningTreeInterfaceAllDetail

counters	*SpanningTreeInterface-Counters* see 2.1757.1.3.3.2	Spanning Tree Interface Counters.
designatedBridgeAddress	*MAC address*	Address of designated bridge.
designatedBridgePriority	*integer*	Priority of designated bridge.
designatedPathCost	*SpanningTreePathCost* see 2.1757.1.3.2.1	Designated path cost.
designatedPortNumber	*integer*	Port number of designated port.
designatedPortPriority	*integer*	Priority of designated port.
designatedRootAddress	*MAC address*	Address of designated root bridge.
designatedRootPriority	*integer*	Priority of designated root bridge.
features	*table(string) = SpanningTreeFeatureStatus* see 2.1757.1.3.3.1	Spanning tree feature status.
forwardDelay	*integer*	Forwarding delay for port (seconds).
maxAge	*integer*	Max Age for port (seconds).
messageAge	*integer*	Message age for port (seconds).
rateLimitEnabled	*boolean*	BPDU rate limiter is enabled.
rateLimitInterval	*floating point number*	BPDU rate limiter window.
rateLimitMaxCount	*integer*	BPDU rate limiter maximum count.
[regionalRootAddress]	*MAC address*	Address of regional root bridge.
[regionalRootPriority]	*integer*	Priority of regional root bridge.
transitionsToForwarding	*integer*	Number of timer port transitioned to forwarding state.

2.1757.1.3.3.1 SpanningTreeFeatureStatus

default	*boolean*	The feature setting is at its default value.
value	*string*	The current feature setting.

2.1757.1.3.3.2 SpanningTreeInterfaceCounters

bpduOtherError	*integer*	BPDUs other error.
bpduRateLimitCount	*integer*	BPDU rate limiter count.
bpduReceived	*integer*	BPDUs received.
bpduSent	*integer*	BPDUs sent.
bpduTaggedError	*integer*	BPDUs tagged error.

2.1757.1.3.4 SpanningTreeInterfaceMstDetail

designatedBridgeAddress	*MAC address*	Address of designated bridge.
designatedBridgePriority	*integer*	Priority of designated bridge.
designatedPathCost	*SpanningTreePathCost* see 2.1757.1.3.2.1	Designated path cost.
designatedPortNumber	*integer*	Port number of designated port.
designatedPortPriority	*integer*	Priority of designated port.
designatedRootAddress	*MAC address*	Address of designated root bridge.
designatedRootPriority	*integer*	Priority of designated root bridge.
[regionalRootAddress]	*MAC address*	Address of regional root bridge.
[regionalRootPriority]	*integer*	Priority of regional root bridge.

2.1757.1.3.5 VlanList

vlans	*list of integer*	A list of vlans.

2.1757.1.4 SpanningTreeRootPort

cost	*SpanningTreePathCost* see 2.1757.1.3.2.1	Root Path Cost.
interface	*interface name*	Root Port Interface Name.
portNumber	*integer*	Port Number of Root Port.

2.1758 show spanning-tree blockedports

spanningTreeInstances	*table(string) = SpanningTreeBlockedPortsList* see 2.1758.1	Map STP instance names to blocked ports.

2.1758.1 SpanningTreeBlockedPortsList

spanningTreeBlockedPorts	*list of Interface name*	List of STP blocked ports.

2.1759 show spanning-tree bridge [...]

[detail]	*SpanningTreeBridgesDetail* see 2.1759.1	Spanning tree bridge details.
spanningTreeBridges	*table(string) = SpanningTreeBridge* see 2.1757.1.1	Map STP bridge names to their details.

2.1759.1 SpanningTreeBridgesDetail

[heartbeatTimeout] **	*floating point number*	Time (in seconds) of no received STP heartbeat before a timeout is declared. **Since: 4.18.0
[lastLocalHeartbeatTimeout] **	*floating point number*	Time of the last local STP heatbeat timeout. **Since: 4.18.0
[lastPeerHeartbeatTimeout] **	*floating point number*	Time of the last peer STP heartbeat timeout. **Since: 4.18.0
[localHeartbeatTimeoutsSinceReboot] **	*integer*	Number of local STP heartbeat timeouts that occured since the last reload. **Since: 4.18.0
mismatchedBpduMode **	*enum: drop | accept*	VLAN-mismatched BPDU handling mode in PVST. **Since: **currentRelease**
[mstPvstBorderEffective]	*boolean*	MST-PVST interoperation is active.
[overriddenBy] **	*string*	By whom Stp is overridden. **Since: 4.18.0
[peerHeartbeatTimeoutsSinceReboot] **	*integer*	Number of peer STP heartbeat timeouts that occured since the last reload. **Since: 4.18.0
[peerHeartbeatWhileEnabled] **	*boolean*	Peer heartbeat check is enabled. **Since: 4.18.0
portChannelPortIdAllocationRange **	*PortChannelPortIdAlloc* see 2.1759.1.1	Allocation range for port-channel port ID. **Since: 4.29.2
[preserveOriginal2p5GAnd5GCost] **	*boolean*	Original 2.5G/5G cost (2000) is preserved. **Since: 4.30.1
restartable	*boolean*	STP is restartable.
[superRoot] **	*boolean*	This switch is network super root. **Since: 4.25.1

2.1759.1.1 PortChannelPortIdAlloc

reservedRangeMax	*integer*	Maximum value for reserved range.
reservedRangeMin	*integer*	Minimum value for reserved range.

2.1760 show spanning-tree bridge assurance [...]

bridgeAssuranceEnabled	*boolean*	Bridge Assurance is enabled in the configuration.
spanningTreeInstances	*table(string) = SpanningTreeBridgeAssuranceInterfaces* see 2.1760.1	Map STP instances to bridge assurance interfaces.

2.1760.1 SpanningTreeBridgeAssuranceInterfaces

interfaces	*table(Interface name) = SpanningTreeBridgeAssuranceValue* see 2.1760.1.1	Map STP interfaces to their Bridge Assurance values.

2.1760.1.1 SpanningTreeBridgeAssuranceValue

bridgeAssuranceValue	*enum: inconsistent | consistent | notNetworkPort*	Spanning Tree Bridge Assurance values.

2.1761 show spanning-tree counters

interfaces	*table(Interface name) = SpanningTreeInterfaceCounters* see 2.1757.1.3.3.2	Map interfaces to STP counters.

2.1762 show spanning-tree instance [...]

see 2.1759

2.1763 show spanning-tree interface ETHINTF [...]

interface	*interface name*	Name of interface.
spanningTreeInstances	*table(string) = SpanningTreeInterface* see 2.1757.1.3	Map STP instances to interfaces.

2.1764 show spanning-tree mst [...]

spanningTreeMstInstances	*table(string) = SpanningTreeMstInstance* see 2.1764.1	Map MST instance names to details.

2.1764.1 SpanningTreeMstInstance

bridge	*SpanningTreeBridgeBrief* see 2.1764.1.1	Spanning tree bridge details.
interfaces	*table(Interface name) = SpanningTreeInterface* see 2.1757.1.3	Map interface names to details.
mappedVlans	*VlanList* see 2.1757.1.3.5	Vlans mapped to the MST instance.
[regionalRootBridge]	*SpanningTreeBridgeBrief* see 2.1764.1.1	Spanning tree regional root bridge details.
[rootBridge]	*SpanningTreeBridgeBrief* see 2.1764.1.1	Spanning tree root bridge details.

2.1764.1.1 SpanningTreeBridgeBrief

Could also be an instance of:
SpanningTreeBridge 2.1757.1.1

macAddress	*MAC address*	MAC address of the bridge.
priority	*integer*	Priority of the bridge (0 to 61440 in increments of 4096).
systemIdExtension	*integer*	System Id extension.

2.1765 show spanning-tree mst configuration [...]

configuredMstInstances	*integer*	Number of MST instances configured.
[mstConfigurationDigest]	*string*	MST configuration digest.
mstConfigurationRevision	*integer*	MST configuration revision.
mstInstances	*table(numeric string) = VlanList* see 2.1757.1.3.5	Map MST instance numbers to corresponding Vlan list.
mstRegionName	*string*	MST region identifier.

2.1766 show spanning-tree mst interface ETHINTF [...]

[spanningTreeMstInterface]	*SpanningTreeMstInterface* see 2.1766.1	Spanning Tree MST interface details.

2.1766.1 SpanningTreeMstInterface

counters	*SpanningTreeInterface-Counters* see 2.1757.1.3.3.2	STP counters.
features	*table(string) = SpanningTreeFeatureStatus* see 2.1757.1.3.3.1	Spanning tree feature status.
interfaceName	*interface name*	Interface Name.
isEdgePort	*boolean*	Interface is an edge port.

`mstInstances`	*table(string)* = *SpanningTreeInterface* see 2.1757.1.3	Map MST instance names to interface details.

2.1767 show spanning-tree mst test information

bridgeAddr	*MAC address*	MAC address of bridge.
mstpRegionId	*string*	MSTP specific region Identifier.
name	*string*	Name of bridge.
stpInstances	*table(string) = BridgeStpiInfo* see 2.1767.1	A mapping of VLAN to spanning tree instances (rapid-pvst) OR the entire group of MSTP instances with key "Mst".
stpVersion	*enum: backup | none | rapidPvstp | rstp | mstp | rapid-pvst | stp*	Version of the protocol being run.

2.1767.1 BridgeStpiInfo

cistPathCost	*integer*	Common Internal Spanning Tree path cost.
cistRoot	*SpanningTreeBridgeBrief* see 2.1764.1.1	Common Internal Spanning Tree root identifier.
interfaces	*table(Interface name) = BridgeIntfInfo* see 2.1767.1.1	A mapping of interface name to its status info.
mstpInstances	*table(string) = BridgeMstiInfo* see 2.1767.1.2	A mapping of Multiple Spanning Tree Instance to its status info.

2.1767.1.1 BridgeIntfInfo

externalPathCost	*integer*	Operational value to use for the external path cost for the port. It is set based on the extPathCost and linkPathCost values in the PortConfig.

2.1767.1.2 BridgeMstiInfo

bridgeId	*SpanningTreeBridgeBrief* see 2.1764.1.1	Bridge identifier on this bridge.
designatedRoot	*SpanningTreeBridgeBrief* see 2.1764.1.1	Designated Root Bridge identifier.
interfaces	*table(Interface name) = BridgeMstiIntfInfo* see 2.1767.1.2.1	A mapping of interface name to its status info.

2.1767.1.2.1 BridgeMstiIntfInfo

fdbFlushes	*integer*	Incremented by one everytime fdb needs to be flushed.
internalPathCost	*integer*	Path cost used on this port, either derived from the link speed or specified via configuration.
portId	*SpanningTreePortId* see 2.1767.1.2.1.1	Port identifier for this MSTI. Each MSTI can have a different port priority resulting in a different port identifier.

role	*enum: backup | master | designated | disabled | root | alternate*	Current port role for this instance and port.

2.1767.1.2.1.1 SpanningTreePortId

number	*integer*	Port number.
priority	*integer*	Priority of port.

2.1768 show spanning-tree root [...]

instances	*table(string) = SpanningTreeRoot* see 2.1768.1	Map STP instance names to their root bridge.

2.1768.1 SpanningTreeRoot

rootBridge	*SpanningTreeBridge* see 2.1757.1.1	Spanning tree root bridge details.
[rootPort]	*SpanningTreeRootPort* see 2.1757.1.4	Root Port.

2.1769 show spanning-tree topology status [...]

topologies	*table(string) = Topology* see 2.1769.1	Maps a topology name to its topology information.
unmappedVlans	*list of integer*	VLAN numbers that are not mapped to any topology.

2.1769.1 Topology

interfaces	*table(string) = PortStatus* see 2.1769.1.1	Maps an interface name to its status.
vlans	*list of integer*	VLANs mapped in this topology.

2.1769.1.1 PortStatus

[lastChange]	*floating point number*	UNIX timestamp with millisecond precision of the last state change for this port.
[numChanges]	*integer*	Number of state changes for this port since the switch booted.
state	*enum: learning | discarding | portStateUnknown | forwarding*	State of the port.

2.1770 show spanning-tree transmit active [...]

see 2.1760

2.1771 show spanning-tree vlan [...]

spanningTreeVlanInstances	*table(numeric string) = SpanningTreeVlanInstance* see 2.1771.1	Map vlans to their STP vlan instances. If a vlan exists, it will belong to this collection regardless of whether there is a spanning tree instance associated with it.

2.1771.1 SpanningTreeVlanInstance

[spanningTreeVlanInstance]	*SpanningTreeInstance* see 2.1757.1	Spanning tree vlan instance details. This field is None when a corresponding vlan exists, but has no spanning tree instance associated with it.

2.1772 show storm-control [...]

[aggregateTrafficClasses]	*table(numeric string) = StormControlType* see 2.1772.1.1	A mapping of Aggregate traffic class to storm control status.
interfaces	*table(Interface name) = IntfStormControlStatus* see 2.1772.1	Per-interface storm control status.

2.1772.1 IntfStormControlStatus

active	*boolean*	Whether storm control has been applied.
errdisabled **	*boolean*	Whether the interface has been errdisabled. **Since: **currentRelease**
reason	*string*	Reason for the status.
trafficTypes	*table(string) = StormControlType* see 2.1772.1.1	Applied storm control setting for a given traffic type.

2.1772.1.1 StormControlType

dormant	*boolean*	Rule is superseded by all traffic.
[drop]	*integer*	Packets dropped by storm control.
level	*integer*	Storm control level.
rate	*integer*	Effective allowed broadcast rate in bps.
thresholdType	*enum: percentage | bitsPerSecond | packetsPerSecond*	Type of threshold applied.

2.1773 show stun client counters [...]

cleanStarts	*integer*	No of times the agent started with proper cleanup done by the previous run.
nonCleanStarts	*integer*	No of times the agent started without proper cleanup done by the previous run.
stunServerProfileCounters	*list of StunServerProfileCountersModel* see 2.1773.1	Counters associated with a server profile.

2.1773.1 StunServerProfileCountersModel

invalidResponses	*integer*	No of invalid responses recevied.
name	*string*	Server profile name.
numRetries	*integer*	No of retries carried out for failed requests.
refreshRequests	*integer*	No of refresh requests sent.
responseProcessingFailures	*integer*	No of times an error was encountered during the processing of a received response.
retriesExhausted	*integer*	No of times the entire retry process failed.
sslTransportCounters	*StunTransportCountersModel* see 2.1773.1.1	SSL transport counters.
successfulResponses	*integer*	No of successful responses received.
tid	*string*	Transaction ID.
udpTransportCounters	*StunTransportCountersModel* see 2.1773.1.1	UDP transport counters.

2.1773.1.1 StunTransportCountersModel

connectionErrors	*integer*	Connection errors.
initializationFailures	*integer*	No of times the initialization failed.
readFailures	*integer*	No of read failures.
successfulInitializations	*integer*	No of successful initializations.
transport	*string*	SSL or UDP.
writeFailures	*integer*	No of write failures.

2.1774 show stun client translations [...]

bindings	*table(string) = StunClientBindingModel* see 2.1774.1	STUN bindings keyed by transaction ID.

2.1774.1 StunClientBindingModel

agentName	*string*	Agent Name.
[lastRefreshed]	*integer*	Seconds elapsed since last response.
[publicAddress]	*IpAddrAndPort* see 2.616.1.1.1.1	Translated IP address and port.
[sourceAddress]	*IpAddrAndPort* see 2.616.1.1.1.1	Source IP address and port.

timeoutInterval	*integer*	STUN bindings timeout interval in seconds.
tlvs	*list of StunTlvModel* see 2.1774.1.1	STUN Attributes.

2.1774.1.1 StunTlvModel

tlvLen	*integer*	STUN TLV length.
tlvType	*string*	STUN TLV type.
tlvValue	*string*	STUN TLV value.

2.1775 show stun server bindings [...]

bindings	*table(string) = StunServerBindingModel* see 2.1775.1	STUN binding bindings keyed by transaction ID.

2.1775.1 StunServerBindingModel

numTlvs	*integer*	Number of STUN Attributes.
publicAddress	*IpAddrAndPort* see 2.616.1.1.1.1	Translated IP address and port.
[timeout]	*integer*	Seconds remaining until timeout.
timeoutInterval	*integer*	Bindings timeout interval in seconds.
tlvs	*list of StunTlvModel* see 2.1774.1.1	STUN Attributes.

2.1776 show stun server status

authMode	*enum: none | ssl | stunLongTermCredentials*	Authentication mode.
bindingTimeout	*integer*	Bindings timeout interval in seconds.
enabled	*boolean*	Service is enabled.
listeningIps	*table(IPv4 or IPv6 address) = Interface name*	Listening IP address and interface of the service.
listeningPort	*integer*	Listening port of the service.
[pid]	*integer*	Process ID of the service.
sslConnectionLifetime	*integer*	SSL connection lifetime in seconds.
timeOfLastRcvdBindingReq	*integer*	Seconds elapsed since the last binding response was received.

2.1777 show switch forwarding-mode

currentL3McastMode **	*enum: storeAndForward | defaultMode | cutThrough*	Current multicast forwarding mode. **Since: 4.21.3
currentMode	*enum: storeAndForward | defaultMode | cutThrough*	Current switching mode.
supportedL3McastModes **	*list of SupportedMode* see 2.1777.1	List of supported multicast modes. **Since: 4.21.3

`supportedModes`	*list of SupportedMode* see 2.1777.1	List of supported modes.

2.1777.1 SupportedMode

`supportedMode`	*enum: storeAndForward* | *defaultMode* | *cutThrough*	Supported switching modes.

2.1778 show switch scheduler

`schedulerMode`	*enum: oversubscribed* | *linerate*	Current scheduler mode.

2.1779 show sync-e esmc counters

clockSourceCounters	*table(Interface name) = SynceEsmcCountersInfo* see 2.1779.1	Clock source RX and TX ESMC counters.

2.1779.1 SynceEsmcCountersInfo

[rxInfo]	*SynceEsmcCountersRxInfo* see 2.1779.1.1	RX ESMC counters information.
[txInfo]	*SynceEsmcCountersTxInfo* see 2.1779.1.2	TX ESMC counters information.

2.1779.1.1 SynceEsmcCountersRxInfo

invalidCount	*integer*	Number of invalid ESMC messages received.
[lastCascadedEecCount] **	*integer*	Last received number of cascaded EECs. **Since: 4.29.1
[lastCascadedEeecCount] **	*integer*	Last received number of cascaded eEECs. **Since: 4.29.1
[lastEssm] **	*integer*	Last received ESSM code. **Since: 4.29.1
[lastQualityLevel]	*enum: QL-PRC | QL-FAILED | QL-EPRC | QL-PRS | QL-ST2 | QL-SMC | QL-SSU-B | QL-INVX | QL-EPRTC | QL-STU | QL-ST3E | QL-EEC1 | QL-UNC | QL-NSUPP | QL-EEC2 | QL-DNU | QL-PROV | QL-ESEC | QL-INV | QL-TNC | QL-DUS | QL-PRTC | QL-SSU-A*	Last received quality level.
[lastSsm]	*integer*	Last received SSM code.
validCount	*integer*	Number of valid ESMC messages received.

2.1779.1.2 SynceEsmcCountersTxInfo

[lastCascadedEecCount] **	*integer*	Last sent number of cascaded EECs. **Since: 4.29.1
[lastCascadedEeecCount] **	*integer*	Last sent number of cascaded eEECs. **Since: 4.29.1
[lastEssm] **	*integer*	Last sent ESSM code. **Since: 4.29.1

[lastQualityLevel]	*enum: QL-PRC* | *QL-FAILED* | *QL-EPRC* | *QL-PRS* | *QL-ST2* | *QL-SMC* | *QL-SSU-B* | *QL-INVX* | *QL-EPRTC* | *QL-STU* | *QL-ST3E* | *QL-EEC1* | *QL-UNC* | *QL-NSUPP* | *QL-EEC2* | *QL-DNU* | *QL-PROV* | *QL-ESEC* | *QL-INV* | *QL-TNC* | *QL-DUS* | *QL-PRTC* | *QL-SSU-A*	Last sent quality level.
[lastSsm]	*integer*	Last sent SSM code.
sentCount	*integer*	Number of ESMC messages sent.

2.1780 show sync-e esmc detail [...]

clockSourceDetail	*table(Interface name) = SynceEsmcDetailInfo* see 2.1780.1	Detailed Clock source RX and TX ESMC information.

2.1780.1 SynceEsmcDetailInfo

[rxInfo]	*SynceEsmcDetailRxInfo* see 2.1780.1.1	Detailed RX ESMC information.
[txInfo]	*SynceEsmcDetailTxInfo* see 2.1780.1.2	Detailed TX ESMC information.

2.1780.1.1 SynceEsmcDetailRxInfo

esmcTypeCount	*table(string) = integer*	ESMC messages received by type.
invalidCount	*integer*	Number of invalid ESMC messages received.
[lastCascadedEecCount] **	*integer*	Last received number of cascaded EECs. **Since: 4.29.1
[lastCascadedEeecCount] **	*integer*	Last received number of cascaded eEECs. **Since: 4.29.1
[lastClockIdentity] **	*string*	Last received Synce clockIdentity of the originator of the extended QlTlv. **Since: 4.29.1
[lastEsmcType]	*enum: information | event*	Last received ESMC message type.
[lastEssm] **	*integer*	Last received ESSM code. **Since: 4.29.1
[lastEtlvFlag] **	*integer*	Last received EEC/eEEC flag. **Since: 4.29.1
[lastInvalidTime]	*floating point number*	UTC timestamp of the last received invalid ESMC message.
[lastQualityLevel]	*enum: QL-PRC | QL-FAILED | QL-EPRC | QL-PRS | QL-ST2 | QL-SMC | QL-SSU-B | QL-INVX | QL-EPRTC | QL-STU | QL-ST3E | QL-EEC1 | QL-UNC | QL-NSUPP | QL-EEC2 | QL-DNU | QL-PROV | QL-ESEC | QL-INV | QL-TNC | QL-DUS | QL-PRTC | QL-SSU-A*	Last received quality level.
[lastSsm]	*integer*	Last received SSM code.
[lastValidTime]	*floating point number*	UTC timestamp of the last received valid ESMC message.
qualityLevelCount	*table(string) = QualityLevelCount* see 2.1780.1.1.1	ESMC messages received by quality level.
[sourceMac]	*MAC address*	Source MAC address of ethernet clock source.

validCount	*integer*	Number of valid ESMC messages received.

2.1780.1.1.1 QualityLevelCount

countBySsm	*table(string) = integer*	ESMC messages by SSM code.

2.1780.1.2 SynceEsmcDetailTxInfo

esmcTypeCount	*table(string) = integer*	ESMC messages sent by type.
[lastCascadedEecCount] **	*integer*	Last sent number of cascaded EECs. **Since: 4.29.1
[lastCascadedEeecCount] **	*integer*	Last sent number of cascaded eEECs. **Since: 4.29.1
[lastClockIdentity] **	*string*	Last sent Synce clockIdentity of the originator of the extended QlTlv. **Since: 4.29.1
[lastEsmcType]	*enum: information | event*	Last sent ESMC message type.
[lastEssm] **	*integer*	Last sent ESSM code. **Since: 4.29.1
[lastEtlvFlag] **	*integer*	Last sent EEC/eEEC flag. **Since: 4.29.1
[lastQualityLevel]	*enum: QL-PRC | QL-FAILED | QL-EPRC | QL-PRS | QL-ST2 | QL-SMC | QL-SSU-B | QL-INVX | QL-EPRTC | QL-STU | QL-ST3E | QL-EEC1 | QL-UNC | QL-NSUPP | QL-EEC2 | QL-DNU | QL-PROV | QL-ESEC | QL-INV | QL-TNC | QL-DUS | QL-PRTC | QL-SSU-A*	Last sent quality level.
[lastSentTime]	*floating point number*	UTC timestamp of the last sent ESMC message.
[lastSsm]	*integer*	Last sent SSM code.
qualityLevelCount	*table(string) = QualityLevelCount* see 2.1780.1.1.1	ESMC messages sent by quality level.
sentCount	*integer*	Number of ESMC messages sent.
[sourceMac]	*MAC address*	Source MAC address of ethernet clock source.

2.1781 show sync-e selection

clockSources	*list of SynceSelectionStatus* see 2.1781.1	List of candidate clock sources participating in the selection process.

2.1781.1 SynceSelectionStatus

clockSource	*ClockSource* see 2.59.1.1	Candidate clock source.
[holdoffTimeout]	*floating point number*	UTC timestamp for the hold-off timer expiry.
priority	*Priority* see 2.1781.1.1	Selection priority.
[qualityLevel]	*enum: QL-PRC* | *QL-FAILED* | *QL-EPRC* | *QL-PRS* | *QL-ST2* | *QL-SMC* | *QL-SSU-B* | *QL-INVX* | *QL-EPRTC* | *QL-STU* | *QL-ST3E* | *QL-EEC1* | *QL-UNC* | *QL-NSUPP* | *QL-EEC2* | *QL-DNU* | *QL-PROV* | *QL-ESEC* | *QL-INV* | *QL-TNC* | *QL-DUS* | *QL-PRTC* | *QL-SSU-A*	Quality level.
status	*string*	Eligibility of clock source.
[waitToRestoreTimeout]	*floating point number*	UTC timestamp for the wait-to-restore timer expiry.

2.1781.1.1 Priority

[disabled]	*boolean*	Clock source will not participate in selection process.
[value]	*integer*	Clock source selection priority.

2.1782 show system bmc status

bmcInReset	*boolean*	BMC is in reset.
[bmcUsb0PingResult]	*enum: bmcInterfaceUndetected* | *pingSuccessful* | *pingUnsuccessful* | *bmcHubInterfaceUndetected*	BMC USB0 IPv6 ping test result.
[personalityOnReboot]	*enum: bmc* | *cpu*	The personality on reboot.

2.1783 show system coredump

coreFiles	*list of string*	List of core files on the device.
mode	*enum: compressedDeferred* | *compressed* | *disabled*	Core dump control mode.

2.1784 show system environment cooling [...]

see 2.319

2.1785 show system environment power

see 2.320

2.1786 show system environment power budget

consumedPower	*floating point number*	Total consumed power in watts.
powerBudget	*floating point number*	Budgeted power in watts.
powerConsumers	*table(string) = PowerConsumer* see 2.1786.1	Devices consuming power.
totalAvailablePower	*floating point number*	Total available power in watts.

2.1786.1 PowerConsumer

amount	*floating point number*	Amount of power the device is consuming in watts.

2.1787 show system environment power current

cardSlots	*table(string) = FruSlot* see 2.322.1	A mapping of labels to card slots.
currentSensors	*table(string) = CurrentSensor* see 2.322.1.1	A mapping of labels to current sensors.

2.1788 show system environment power detail

see 2.320

2.1789 show system environment power histogram

buckets	*table(numeric string) = integer*	Mapping of power ranges (in percent) to sample counts.
sampleRate	*floating point number*	The rate (in samples per second) at which samples are collected.

2.1790 show system environment power voltage

revision 2

see 2.322

2.1791 show system environment temperature [...]

see 2.323

2.1792 show system environment temperature transceiver [...]

see 2.324

2.1793 show system health storage

devices	*table(string) = StorageHealth* see 2.1793.1	Storage devices.

2.1793.1 StorageHealth

[emmc]	*EmmcLifetime* see 2.1793.1.1	eMMC lifetimes attributes.
[smart]	*SmartHealth* see 2.1793.1.2	S.M.A.R.T. health attributes.

2.1793.1.1 EmmcLifetime

[mlcWearLevel]	*floating point number*	MLC region fractional design wear remaining.
[reservesLevel]	*floating point number*	Reserve block fractional level remaining.
[slcWearLevel]	*floating point number*	SLC region fractional design wear remaining.

2.1793.1.2 SmartHealth

[health]	*string*	Overall device health.
[reallocationsRemaining]	*floating point number*	Fractional unused reallocations blocks.
[wearLevel]	*floating point number*	Percent design lifetime wear remaining.

2.1794 show system storage health

see 2.1793

2.1795 show system storage health smart

devices	*table(string) = DeviceSmartAttrs* see 2.1795.1	SMART attributes, keyed by device name.

2.1795.1 DeviceSmartAttrs

attributes	*list of SmartAttr* see 2.1795.1.1	SMART attribute list.

2.1795.1.1 SmartAttr

name	*string*	SMART attribute name.
status	*enum: n/a | ok | failed*	SMART attribute status.
[unit]	*string*	Units for value.
[value]	*integer*	SMART attribute value.

2.1796 show system storage health smart detail

revision 2

devices	*table(string) = DeviceSmartAttrDetails* see 2.1796.1	SMART attributes, keyed by device.

2.1796.1 DeviceSmartAttrDetails

attributes	*list of SmartAttrDetail* see 2.1796.1.1	SMART attribute list.

2.1796.1.1 SmartAttrDetail

[id]	*integer*	SMART attribute ID.
name	*string*	SMART attribute name.
[normalized]	*integer*	SMART normalized value.
raw	*integer*	SMART raw value.
[threshold]	*integer*	SMART threshold (lowest acceptable).
[worst]	*integer*	SMART worst normalized value.

2.1797 show system storage statistics

devices	*table(string) = IOStat* see 2.1797.1	Storage devices.

2.1797.1 IOStat

measurementTimestamp	*floating point number*	Timestamp at which last measurement was done.
measurementType	*enum: boot | total | lastDay | dayAverage*	Type and length of measurement.
read	*integer*	Quantity of bytes read.
write	*integer*	Quantity of bytes written.

2.1798 show system storage statistics last day

see 2.1797

2.1799 show system storage statistics rates

see 2.1797

2.1800 show system systemd [...]

unitsStatus	*table(string) = SystemdUnitStatus* see 2.1800.1	Mapping of a systemd unit to its status.

2.1800.1 SystemdUnitStatus

unitDesc	*string*	Brief description of unit.
unitState	*enum: remounting | start-post | mounted | mounting-sigterm | listening | stop | remounting-sigterm | start | stop-sigterm | undefined | remounting-sigkill | stop-post | running | mounting-done | start-chown | stop-pre-sigterm | failed | stop-sigabrt | mounting-sigkill | exited | start-pre | final-sigkill | unmounting | unmounting-sigterm | reload | dead | stop-sigkill | final-sigterm | unmounting-sigkill | stop-pre-sigkill | mounting | auto-restart*	Active state of a systemd unit.

2.1801 show system systemd detail

systemdMounts	*SystemdMountpointsDetail* see 2.1801.1	Systemd aware mount points.
systemdServices	*SystemdServicesDetail* see 2.1801.2	Systemd managed services information.
systemdSockets	*SystemdSocketsDetail* see 2.1801.3	Systemd managed sockets information.

2.1801.1 SystemdMountpointsDetail

systemdMountpointsInfo	*table(string) = SystemdMountpointInfo* see 2.1801.1.1	Mapping of a systemd mount to detailed-description of its state.

2.1801.1.1 SystemdMountpointInfo

fsType	*string*	Filesystem type.
mountDesc	*string*	Mount point description.
mountPath	*string*	Where is the mountpoint.
mountState	*enum: remounting* | *start-post* | *mounted* | *mounting-sigterm* | *listening* | *stop* | *remounting-sigterm* | *start* | *stop-sigterm* | *undefined* | *remounting-sigkill* | *stop-post* | *running* | *mounting-done* | *start-chown* | *stop-pre-sigterm* | *failed* | *stop-sigabrt* | *mounting-sigkill* | *exited* | *start-pre* | *final-sigkill* | *unmounting* | *unmounting-sigterm* | *reload* | *dead* | *stop-sigkill* | *final-sigterm* | *unmounting-sigkill* | *stop-pre-sigkill* | *mounting* | *auto-restart*	Active state of a systemd unit.
[mountTime]	*floating point number*	Timestamp of when unit reached current state.

2.1801.2 SystemdServicesDetail

systemdServicesInfo	*table(string) = SystemdServiceData* see 2.1801.2.1	Mapping of a systemd service unit to de-tailed description of its state.

2.1801.2.1 SystemdServiceData

cgrpName	*string*	Name of with CGroup for system service unit.
cgrpTasks	*table(numeric string) = string*	Mapping of a cgroup process id to its command line.
svcActvState	*enum: remounting | start-post | mounted | mounting-sigterm | listening | stop | remounting-sigterm | start | stop-sigterm | undefined | remounting-sigkill | stop-post | running | mounting-done | start-chown | stop-pre-sigterm | failed | stop-sigabrt | mounting-sigkill | exited | start-pre | final-sigkill | unmounting | unmounting-sigterm | reload | dead | stop-sigkill | final-sigterm | unmounting-sigkill | stop-pre-sigkill | mounting | auto-restart*	Active state of a systemd unit.
svcDesc	*string*	Systemd service short description.
svcPid	*integer*	Process id. Not shown for reloaded SYS-V service.
svcState	*string*	Running/Exited status of service with status code.
[timestamp]	*floating point number*	Timestamp of when unit reached current state.

2.1801.3 SystemdSocketsDetail

systemdSocketsInfo	*table(string) = SystemdSocketInfo* see 2.1801.3.1	Mapping of a systemd socket to detailed-description of its state.

2.1801.3.1 SystemdSocketInfo

sockDesc	*string*	Description of socket.
sockState	*enum: remounting* | *start-post* | *mounted* | *mounting-sigterm* | *listening* | *stop* | *remounting-sigterm* | *start* | *stop-sigterm* | *undefined* | *remounting-sigkill* | *stop-post* | *running* | *mounting-done* | *start-chown* | *stop-pre-sigterm* | *failed* | *stop-sigabrt* | *mounting-sigkill* | *exited* | *start-pre* | *final-sigkill* | *unmounting* | *unmounting-sigterm* | *reload* | *dead* | *stop-sigkill* | *final-sigterm* | *unmounting-sigkill* | *stop-pre-sigkill* | *mounting* | *auto-restart*	Active state of a systemd unit.
sockType	*string*	Type of socket ie., stream/datagram etc.
[timestamp]	*floating point number*	Timestamp of when unit reached current state.

2.1802 show system systemd mountpoints detail

see 2.1801.1

2.1803 show system systemd services detail

see 2.1801.2

2.1804 show system systemd sockets detail

see 2.1801.3

2.1805 show tacacs

[groups]	*table(string) = ServerGroup* see 2.1805.1	Mapping of group name to group information.
lastCounterClearTime	*floating point number*	UTC time when the counters were last cleared.
srcIntf	*table(string) = Interface name*	Configured source-interface name.
tacacsServers	*list of TacacsStats* see 2.1805.2	Mapping of server IP address or hostname to server information.

2.1805.1 ServerGroup

members	*list of ServerInfo* see 2.1805.1.1	Mapping of server IP address or hostname to group information.
serverGroup	*string*	Configured server-group.

2.1805.1.1 ServerInfo

authport	*integer*	Specifies server authorization port number.
hostname	*string*	IP address or hostname of the TACACS+ server.
[vrf] **	*string*	The VRF over which this server is reachable. **Since: 4.25.1

2.1805.2 TacacsStats

connectionCloses	*integer*	Number of successfully closed TCP connections.
connectionDisconnects	*integer*	Number of disconnected TCP connections to the TACACS+ server.
connectionFailures	*integer*	Number of failed TCP connections to the TACACS+ server.
connectionOpens	*integer*	Number of successful TCP connections to the TACACS+ server.
connectionTimeouts	*integer*	Number of timedout TCP connections to the TACACS+ server.
dnsErrors **	*integer*	Number of DNS errors by the TACACS+ server. **Since: 4.25.1
messagesReceived	*integer*	Number of packets received from the TACACS+ server.
messagesSent	*integer*	Number of packets sent to the TACACS+ server.
receiveErrors	*integer*	Number of errors received from the TACACS+ server.
receiveTimeouts	*integer*	Number of timeouts received from the TACACS+ server.
sendTimeouts	*integer*	Number of timeouts sent to the TACACS+ server.

serverInfo	*ServerInfo* see 2.1805.1.1	Configured server information.
unknownMandatoryAttr-Failures **	*integer*	Number of unknown mandatory attributes received from the TACACS+ server which failed the authorizations. **Since: 4.29.1
unknownMandatoryAttrIgnored **	*integer*	Number of unknown mandatory attributes received from the TACACS+ server which got ignored. **Since: 4.29.1

2.1806 show tap aggregation counters default-forwarding

interfaces	*table(Interface name) = CounterValue* see 2.1806.1	A mapping from interfaces to counter values.

2.1806.1 CounterValue

byteCount	*integer*	The number of bytes that have been counted.
packetCount	*integer*	The number of packets that have been counted.

2.1807 show tap aggregation counters tunnel termination gre

tunnels	*table(string) = CounterValue* see 2.1806.1	A mapping from TAP tunnels to counter values.

2.1808 show tap aggregation groups [...]

groups	*table(string) = TapAggGroup* see 2.1808.1	Tap-agg groups keyed by group name.

2.1808.1 TapAggGroup

tapPortStatuses	*table(Interface name) = PortStatus* see 2.1808.1.1	Mapping from interface name to a list of theirstatuses and configured policies.
toolPortStatuses	*table(Interface name) = PortStatus* see 2.1808.1.1	Mapping from interface name to a list of theirstatuses.

2.1808.1.1 PortStatus

policies **	*list of string*	List policies by whichthe port is added to the group. **Since: 4.18.1
statuses	*list of string*	List statuses of the interface. Active, Interface status not found, Link down, Link unknown, Lag member, Mirror destination.

2.1809 show tap aggregation groups map TAPPORT [...]

tapPortsMap	*table(Interface name) = MapPropertyList* see 2.1809.1	A mapping from tap to tool ports.

2.1809.1 MapPropertyList

mapPropertyList	*list of MapProperty* see 2.1809.1.1	List of tap/tool map properties.

2.1809.1.1 MapProperty

groupName	*string*	Group Name.
mappedIntfName	*interface name*	Tap/Tool Port Name.
policyName	*string*	PolicyMap/ClassMap Name.

2.1810 show tap aggregation groups map TOOLPORT [...]

toolPortsMap	*table(Interface name) = MapPropertyList* see 2.1809.1	A mapping from tool to tap ports.

2.1811 show tcpdump

revision 2

sessions	*table(string) = SessionModel* see 2.1811.1	A mapping between session name and session information.

2.1811.1 SessionModel

duration	*integer*	Capture duration (seconds).
enabled	*boolean*	Enabled.
errorMessage	*string*	Error message associated with the session.
interface	*string*	Capture source.
maxFileSize	*integer*	Max file size (millions of bytes).
maxPerPacketCaptureSize	*integer*	Per-packet capture size (bytes).
outputFile	*string*	Output file.
packetCount	*integer*	Packet count.

packetFilter	*string*	Packet filter.
running	*boolean*	Indicates if capture is running.
vrfName	*string*	VRF name.

2.1812 show terminal

length	*integer*	Length of the terminal in rows.
width	*integer*	Width of the terminal in columns.

2.1813 show trace AGENT

configTrace	*list of string*	Trace setting for agent when agent is not running.
[destination]	*string*	File name where trace output goes if not using the standard log file.
execTrace **	*list of string*	Active exec trace settings for agent. **Since: 4.29.2
execTraceStartTime **	*integer*	Time at which the exec trace was initialized. **Since: 4.29.2
execTraceTimeout **	*integer*	Time remaining in exec traces. **Since: 4.29.1
facilities	*table(string) = StateAndLevel* see 2.1813.1	A mapping of facility name to its trace settings.
[facilityNameWidth]	*integer*	Set the width of facility name in the trace output.
[filenameWidth]	*integer*	Set the width of file name in the trace output.
[functionNameWidth]	*integer*	Set the width of function name in the trace output.
showDate	*boolean*	Date in trace messages.
showFacilityName	*boolean*	Facility name in trace messages.
showFileName	*boolean*	File name in trace messages.
showFunctionName	*boolean*	Method name in trace messages.
showLineNumber	*boolean*	Line number in trace messages.
showPid	*boolean*	Pid in trace messages.
showTid	*boolean*	Tid in trace messages.
showTraceTimestamp	*boolean*	Timestamp in trace messages.
showTracelevel	*boolean*	Trace level in trace messages.

2.1813.1 StateAndLevel

levels	*list of string*	List of levels for which tracing has been enabled.

2.1814 show traffic-engineering cbf fec [...]

vrfs	*table(string) = CbfFecModel* see 2.1814.1	Per VRF Traffic Engineering CBF override entries.

2.1814.1 CbfFecModel

overrides	*list of CbfFecOverride* see 2.1814.1.1	CBF FEC override entry.

2.1814.1.1 CbfFecOverride

defaultFecId	*integer*	Default FEC ID.
[dscp]	*integer*	DSCP value.
ipVersion	*enum: ipv4 | ipv6*	IP version.
overrideFecId	*integer*	Overriding SR-TE Policy FEC ID.
[tc] **	*integer*	Traffic class value. **Since: 4.29.0
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Nexthop | IS-IS SR | EosSdk sBFD | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | RSVP FRR | RSVP LER | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel Types.

2.1815 show traffic-engineering cspf path IP_ADDR links

vrfs	*table(string) = CspfPathLinksVrfModel* see 2.1815.1	A mapping between VRF and information of all CSPF paths in that VRF.

2.1815.1 CspfPathLinksVrfModel

[v4Info]	*CspfPathLinksAfModel* see 2.1815.1.1	CSPF path information for IPv4 address family.
[v6Info]	*CspfPathLinksAfModel* see 2.1815.1.1	CSPF path information for IPv6 address family.

2.1815.1.1 CspfPathLinksAfModel

pathIps	*table(IPv4 or IPv6 address) = CspfPathLinksDestIpModel* see 2.1815.1.1.1	A mapping of a destination IP to all CSPF paths for that destination IP.

2.1815.1.1.1 CspfPathLinksDestIpModel

paths	*table(numeric string) = CspfPathLinksEntryModel* see 2.1815.1.1.1.1	A mapping of an id representing a constraint to the corresponding CSPF path.

2.1815.1.1.1.1 CspfPathLinksEntryModel

destinationOnSourceNode	*boolean*	Destination IP address is on the source node.
[links]	*table(numeric string) = CspfPathLinkAttrsModel* see 2.1815.1.1.1.2	TE information of each link in the CSPF path.
status	*enum: pathNotFound | cspfPending | pathFound*	Status of path computation.

2.1815.1.1.1.2 CspfPathLinkAttrsModel

found	*boolean*	Link found in topology.
[hopAddress]	*IP address*	Next hop address.
[linkAttributes]	*LinkAttributesModel* see 2.1815.1.1.1.3	Link attributes.

2.1815.1.1.1.3 LinkAttributesModel

[administrativeGroup]	*integer*	Administrative Group of the Link.
destinationIgpRouterId	*string*	Remote IGP Router ID.
destinationTeRouterId	*IP address*	Remote TE Router ID.
interfaceAddresses	*list of InterfaceAddress* see 2.1815.1.1.1.4	List of local Interface Addresses forming the adjacency.
[maxLinkBw]	*integer*	Maximum bandwidth (bps) that can be used on Directed Link.

[maxReservableBw]	*integer*	Maximum bandwidth (bps) that can be reserved on Directed Link.
[metric]	*integer*	TE Link cost.
neighborAddresses	*list of InterfaceAddress* see 2.1815.1.1.1.4	List of remote Neighbor Addresses forming the adjacency.
networkType	*enum: p2p | lan*	Network type.
[sharedRiskLinkGroups]	*list of SharedRiskLinkGroupWithCost* see 2.1815.1.1.1.6	List of Shared Risk Link Groups.
sourceIgpRouterId	*string*	Local IGP Router ID.
sourceTeRouterId	*IP address*	Local TE Router ID.
topology	*TopologyModel* see 2.1815.1.1.1.7	Topology type information.
[unreservedBw]	*ReservablePriorityBandwidth* see 2.1815.1.1.1.5	Maximum bandwidth reservable for a priority.

2.1815.1.1.1.4 InterfaceAddress

[ipv4Address]	*IPv4 address*	IPv4 address.
[ipv6Address]	*IPv6 address*	IPv6 address.

2.1815.1.1.1.5 ReservablePriorityBandwidth

priority0	*integer*	Reservable Bandwidth (bps) for priority level 0.
priority1	*integer*	Reservable Bandwidth (bps) for priority level 1.
priority2	*integer*	Reservable Bandwidth (bps) for priority level 2.
priority3	*integer*	Reservable Bandwidth (bps) for priority level 3.
priority4	*integer*	Reservable Bandwidth (bps) for priority level 4.
priority5	*integer*	Reservable Bandwidth (bps) for priority level 5.
priority6	*integer*	Reservable Bandwidth (bps) for priority level 6.
priority7	*integer*	Reservable Bandwidth (bps) for priority level 7.

2.1815.1.1.1.6 SharedRiskLinkGroupWithCost

cost	*integer*	Cost of SRLG Group.
groupId	*integer*	Shared Risk Link Group Identifier.
[groupName]	*string*	Shared Risk Link Group Name.

2.1815.1.1.1.7 TopologyModel

[isis]	*IsisTopologyInfoModel* see 2.1815.1.1.1.8	IS-IS topology information.
[ospf]	*OspfTopologyInfoModel* see 2.1815.1.1.1.9	OSPF topology information.

2.1815.1.1.1.8 IsisTopologyInfoModel

level	*integer*	IS-IS level.

2.1815.1.1.1.9 OspfTopologyInfoModel

areaId	*IPv4 address*	OSPF area ID.
instanceId	*integer*	OSPF instance ID.
version	*integer*	OSPF version.

2.1816 show traffic-engineering cspf path PATH_ID [...]

vrfs	*table(string) = CspfPathByIdVrfModel* see 2.1816.1	A mapping between Vrf and information of all CSPF paths in that vrf.

2.1816.1 CspfPathByIdVrfModel

[v4Info]	*CspfPathByIdEntryModel* see 2.1816.1.1	CSPF path information for IPv4 address family.
[v6Info]	*CspfPathByIdEntryModel* see 2.1816.1.1	CSPF path information for IPv6 address family.

2.1816.1.1 CspfPathByIdEntryModel

constraint	*CspfPathConstraintModel* see 2.1816.1.1.1	Path constraint.
[details]	*Details* see 2.1816.1.1.3	Detailed path information.
[dstAddr]	*IP address*	Address for destination.
hops	*list of CspfPathHopModel* see 2.1816.1.1.2	Hops specifying the path to the destination.
[pathId]	*integer*	Path ID.
status	*enum: pathNotFound | cspfPending | pathFound*	Status of path computation.

2.1816.1.1.1 CspfPathConstraintModel

[bandwidth]	*integer*	Bandwidth reserved for this path in bits per second.
[bwSetupPriority]	*integer*	Priority of bandwidth reservation request.
[excludeAdminGroup]	*integer*	Admin Group excluded.
[excludeIntf]	*interface name*	Interface excluded from this path.
[excludeLinksWithAddress]	*list of IP address*	Addresses of links excluded.
[excludeNode]	*string*	Node excluded from this path.
[excludeSrlgOfIntf]	*IntfSrlg* see 2.804.1.1.1.6	SRLG excluded of interface.
[excludeSrlgOfPaths] **	*list of CspfPathHelperModel* see 2.1816.1.1.1.2	List of all paths whose SRLGs to be excluded. **Since: 4.25.1
[explicitPath]	*list of IP address*	Complete list of hops making up this path.
[includeAllAdminGroup]	*integer*	Admin Group included.
[includeAnyAdminGroup]	*integer*	Any Admin Group included.
[includeHops]	*list of CspfIncludeHopModel* see 2.1816.1.1.1.1	Nodes or links included in this path.
[sharedBwGroupId]	*integer*	Shared bandwidth group ID.

2.1816.1.1.1.1 CspfIncludeHopModel

hop	*IP address*	Interface address or TE Router ID of the hop.
loose	*boolean*	The include hop constraint is loose.

2.1816.1.1.1.2 CspfPathHelperModel

destination	*IP address*	Destination address.
pathId	*integer*	Path index.
[srlgIds]	*list of integer*	List of SRLG IDs for the path.

2.1816.1.1.2 CspfPathHopModel

[includeIps]	*list of IP address*	Non-ingress IP addresses in includeHop constraint corresponding to teRouterId.
ipAddr	*IP address*	Interface IP address.
teRouterId	*IP address*	TE Router ID.

2.1816.1.1.3 Details

autoUpdate	*boolean*	Automatically re-optimize the path.
[bwLocallyAccounted]	*boolean*	Bandwidth is locally accounted as reserved.
[changeCount]	*integer*	Number of times path updated.
[lastUpdatedTime]	*floating point number*	UTC timestamp of the last path update.
refreshReqSeq	*integer*	Request sequence number.
[refreshRespSeq]	*integer*	Response sequence number.

2.1817 show traffic-engineering cspf path PATH_ID links

vrfs	*table(string) = CspfPathByIdLinksVrfModel* see 2.1817.1	A mapping between VRF and information of all CSPF paths in that VRF.

2.1817.1 CspfPathByIdLinksVrfModel

[v4Info]	*CspfPathByIdLinksAfModel* see 2.1817.1.1	CSPF path information for IPv4 address family.
[v6Info]	*CspfPathByIdLinksAfModel* see 2.1817.1.1	CSPF path information for IPv6 address family.

2.1817.1.1 CspfPathByIdLinksAfModel

destinationOnSourceNode	*boolean*	Destination IP address is on the source node.
dstAddr	*IP address*	Address for destination.
[links]	*table(numeric string) = CspfPathLinkAttrsModel* see 2.1815.1.1.1.2	TE information of each link in the CSPF path.

pathId	*integer*	Path ID.
status	*enum: pathNotFound | cspfPending | pathFound*	Status of path computation.

2.1818 show traffic-engineering cspf path [...]

revision 2

vrfs	*table(string) = CspfPathVrfModel* see 2.1818.1	A mapping between Vrf and information of all CSPF paths in that vrf.

2.1818.1 CspfPathVrfModel

[v4Info]	*CspfPathAfModel* see 2.1818.1.1	CSPF path information for IPv4 address family.
[v6Info]	*CspfPathAfModel* see 2.1818.1.1	CSPF path information for IPv6 address family.

2.1818.1.1 CspfPathAfModel

pathIps	*table(IPv4 or IPv6 address) = CspfPathDestIpModel* see 2.1818.1.1.1	A mapping of a destination IP to all CSPF paths for that destination IP.

2.1818.1.1.1 CspfPathDestIpModel

paths	*table(numeric string) = CspfPathEntryModel* see 2.1818.1.1.1.1	A mapping of an id representing a constraint to the corresponding CSPF path.

2.1818.1.1.1.1 CspfPathEntryModel

constraint	*CspfPathConstraintModel* see 2.1816.1.1.1	Path constraint.
[details]	*Details* see 2.1818.1.1.1.2	Detailed path information.
hops	*list of CspfPathHopModel* see 2.1816.1.1.2	Hops specifying the path to the destination.
status	*enum: pathNotFound | cspfPending | pathFound*	Status of path computation.

2.1818.1.1.1.2 Details

autoUpdate	*boolean*	Automatically re-optimize the path.
[bwLocallyAccounted] **	*boolean*	Bandwidth is locally accounted as reserved. **Since: 4.28.1
[changeCount]	*integer*	Number of times path updated.
[lastUpdatedTime]	*floating point number*	UTC timestamp of the last path update.
refreshReqSeq	*integer*	Request sequence number.
[refreshRespSeq]	*integer*	Response sequence number.

2.1819 show traffic-engineering database

revision 2

vrfs	*table(string) = TeVrfDbModel* see 2.1819.1	TE DB information per vrf.

2.1819.1 TeVrfDbModel

[ipv4]	*TeAfDbModel* see 2.1819.1.1	TE DB information for IPv4 address family.
[ipv6]	*TeAfDbModel* see 2.1819.1.1	TE DB information for IPv6 address family.

2.1819.1.1 TeAfDbModel

teRouterIds	*table(string) = TeRouterDbModel* see 2.1819.1.1.1	TE DB information per TE Router ID.

2.1819.1.1.1 TeRouterDbModel

igps	*table(string) = IgpInstanceInfoModel* see 2.1819.1.1.1.1	A mapping of the IGP name to the IGP information.

2.1819.1.1.1.1 IgpInstanceInfoModel

instances	*table(numeric string) = IgpModel* see 2.1819.1.1.1.2	A mapping of the IGP instance ID to the IGP database.

2.1819.1.1.1.2 IgpModel

areas	*table(IPv4 address) = TeSourceProtocolDbModel* see 2.1819.1.1.1.3	A mapping of the OSPF area ID to the area's TE database information.
levels	*table(numeric string) = TeSourceProtocolDbModel* see 2.1819.1.1.1.3	A mapping of the IS-IS level to the corresponding IS-IS TE database.

2.1819.1.1.1.3 TeSourceProtocolDbModel

routers	*table(string) = RouterInfoModel* see 2.1819.1.1.1.4	Protcol specific router information.

2.1819.1.1.1.4 RouterInfoModel

[lan]	*TeLinksModel* see 2.1819.1.1.1.5	TE DB information for LAN links.
numLinks	*integer*	Number of links.

[p2p]	*TeLinksModel* see 2.1819.1.1.1.5	TE DB information for P2P links.

2.1819.1.1.1.5 TeLinksModel

links	*list of NgbTeInfoModel* see 2.1819.1.1.1.6	TE information of each neighboring link.

2.1819.1.1.1.6 NgbTeInfoModel

[administrativeGroup]	*integer*	Administrative Group of the Link.
interfaceAddresses	*list of InterfaceAddress* see 2.1815.1.1.1.4	List of local Interface Addresses forming the adjacency.
[maxLinkBw]	*integer*	Maximum bandwidth (bps) that can be usedon Directed Link.
[maxReservableBw]	*integer*	Maximum bandwidth (bps) that can bereserved on Directed Link.
[metric]	*integer*	TE Link cost.
[neighborAddresses]	*list of InterfaceAddress* see 2.1815.1.1.1.4	List of remote Neighbor Addresses forming the adjacency.
neighborId	*string*	Neighbor Protocol ID.
[sharedRiskLinkGroups]	*list of SharedRiskLinkGroup* see 2.1819.1.1.1.7	List of Shared Risk Link Group.
[unreservedBw]	*ReservablePriority-Bandwidth* see 2.1815.1.1.1.5	Maximum bandwidth reservable for a priority.

2.1819.1.1.1.7 SharedRiskLinkGroup

Could also be an instance of:
SharedRiskLinkGroupWithCost 2.1815.1.1.1.6

groupId	*integer*	Shared Risk Link Group Identifier.
[groupName]	*string*	Shared Risk Link Group Name.

2.1820 show traffic-engineering database duplicate interface-address

intfIps	*table(IPv4 or IPv6 address) = InterfaceIpInfoModel* see 2.1820.1	TE information of each duplicate interface IP.

2.1820.1 InterfaceIpInfoModel

teRouterIds	*list of IP address*	TE RouterIds of the routers originating the IP address.

2.1821 show traffic-engineering database router-id IP

revision 2

see 2.1819

2.1822 show traffic-engineering gribi ipv4-aft [...]

vrfs	*table(string) = GribiIpAft* see 2.1822.1	IP AFT entries per VRF.

2.1822.1 GribiIpAft

prefixes	*table(IPv4 or IPv6 address prefix) = GribiIpAftEntry* see 2.1822.1.1	AFT entries keyed by prefix.

2.1822.1.1 GribiIpAftEntry

[fibAckAcknowledged] **	*AcknowledgementRouteVersionId* see 2.1822.1.1.1	Acknowledged FIB-ACK route version ID. **Since: 4.28.2
[fibAckRequested] **	*AcknowledgementRouteVersionId* see 2.1822.1.1.1	Requested FIB-ACK route version ID. **Since: 4.27.1
[metadata] **	*string*	Metadata. **Since: 4.27.1
nhgId	*integer*	Nexthop Group ID.
prefix	*IP address prefix*	Prefix.

2.1822.1.1.1 AcknowledgementRouteVersionId

failed **	*boolean*	Route failed installation in hardware. **Since: 4.28.2
requestedDelete	*boolean*	Route due to be deleted.
versionId	*integer*	Route version ID to be acknowledged.

2.1823 show traffic-engineering gribi ipv4-aft unacknowledged [...]

vrfs	*table(string) = GribiPendingIpAft* see 2.1823.1	Unacknowledged AFT entries per VRF.

2.1823.1 GribiPendingIpAft

unacknowledgedPrefixes	*table(string) = GribiIpAftVersionType* see 2.1823.1.1	Map of versioned Route AFT entries, keyed by Prefix.

2.1823.1.1 GribiIpAftVersionType

ipVersionAft	*table(numeric string) = GribiIpAftEntry* see 2.1822.1.1	Map of Route AFT entries, keyed by version ID.

2.1824 show traffic-engineering gribi ipv6-aft [...]

see 2.1822

2.1825 show traffic-engineering gribi ipv6-aft unacknowledged [...]

see 2.1823

2.1826 show traffic-engineering gribi mpls-aft [...]

mplsAft	*table(numeric string) = GribiMplsAftType* see 2.1826.1	Map of MPLS AFT entries, keyed by Label.

2.1826.1 GribiMplsAftType

label	*integer*	Label.
[mplsAftDetail]	*GribiMplsAftDetail* see 2.1826.1.1	gRIBI MPLS AFT detail.
nhgId	*integer*	Nexthop Group ID.

2.1826.1.1 GribiMplsAftDetail

[fecDetail]	*GribiFecDetail* see 2.1826.1.1.1	gRIBI FEC detail.

2.1826.1.1.1 GribiFecDetail

fecId	*integer*	FEC ID.
[fecVias]	*list of GribiFecVia* see 2.1826.1.1.1.1	List of FEC vias.

2.1826.1.1.1.1 GribiFecVia

[mplsVias]	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	Segment Routing Traffic Engineering policy information.
tunnelIdx	*integer*	Tunnel index.
[weight]	*integer*	Weight of the nexthop.

2.1826.1.1.1.2 SrTeSegmentListVia

interface	*interface name*	Egress L3 interface of the next hop.
[mplsLabels]	*list of integer*	MPLS label stack.
nexthop	*IP address*	Next hop IP address.

2.1827 show traffic-engineering gribi nexthop-aft [...]

revision 2

nhAft	*table(numeric string) = GribiNhAftType* see 2.1827.1	Map of Nexthop AFT entries, keyed by Nexthop ID.

2.1827.1 GribiNhAftType

[decapActionType]	*ipip*	Decapsulation type.
[encapActionType]	*ipip*	Encapsulation type.
[interface]	*interface name*	Next hop interface.
[ipAddress]	*IP address*	Next hop address.
[ipInIpEncap]	*GribiIpInIpEncap* see 2.1827.1.1	IpInIp Encapsulation.
[macAddress]	*MAC address*	Next hop MAC address.
[nhAftDetail]	*GribiNhAftDetail* see 2.1827.1.2	gRIBI Nexthop AFT detail.
nhId	*integer*	Nexthop ID.
pushMplsLabels	*list of integer*	Push MPLS Label Stack Data.
[vrf]	*string*	VRF name.

2.1827.1.1 GribiIpInIpEncap

dstIp	*IP address*	Destination address.
srcIp	*IP address*	Source address.

2.1827.1.2 GribiNhAftDetail

[tunnelDetail]	*GribiFecVia* see 2.1826.1.1.1.1	Tunnel detail.

2.1828 show traffic-engineering gribi nexthop-aft unacknowledged [...]

unacknowledgedNhs	*table(numeric string) = GribiNhAftVersionType* see 2.1828.1	Map of versioned Nexthop AFT entries, keyed by Nexthop ID.

2.1828.1 GribiNhAftVersionType

nhVersionAft	*table(numeric string) = GribiNhAftType* see 2.1827.1	Map of Nexthop AFT entries, keyed by versionId.

2.1829 show traffic-engineering gribi nexthop-group-aft [...]

details_	*boolean*	gRIBI Nexthop Group AFT Detail.
nhgAft	*table(numeric string) = GribiNhgAftType* see 2.1829.1	Map of Nexthop Group AFT entries, keyed by Nexthop Group ID.

2.1829.1 GribiNhgAftType

[backupNhgId] **	*integer*	Backup nexthop group ID. **Since: 4.28.1
[fecId]	*integer*	FEC ID.
[fibAckAcknowledged] **	*AcknowledgementFecVersionId* see 2.1829.1.1	Acknowledged FIB-ACK FEC version ID. **Since: 4.28.0
[fibAckRequested] **	*AcknowledgementFecVersionId* see 2.1829.1.1	Requested FIB-ACK FEC version ID. **Since: 4.27.2
nhgId	*integer*	Nexthop Group ID.
nhgNhs	*list of GribiNhgNhAftType* see 2.1829.1.2	List of nexthops in a Nexthop Group AFT entry.

2.1829.1.1 AcknowledgementFecVersionId

failed **	*boolean*	FEC install failed. **Since: 4.28.2
requestedDelete	*boolean*	FEC due to be deleted.
versionId	*integer*	FEC version ID to be acknowledged.

2.1829.1.2 GribiNhgNhAftType

[decapActionType] **	*ipip*	Decapsulation type. **Since: 4.28.1
[encapActionType] **	*ipip*	Encapsulation type. **Since: 4.28.1
[fallbackVrf] **	*string*	Fallback VRF name. **Since: 4.28.1
[interface] **	*interface name*	Next hop interface. **Since: 4.26.2
[ipAddress] **	*IP address*	Next hop address. **Since: 4.26.2
[ipInIpEncap] **	*GribiIpInIpEncap* see 2.1827.1.1	IpInIp Encapsulation. **Since: 4.28.1
[macAddress] **	*MAC address*	Next hop MAC address. **Since: 4.28.1
[mplsVias]	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	Segment Routing Traffic Engineering policy information.
nhId	*integer*	Nexthop ID.
weight	*integer*	Weight of the nexthop.

2.1830 show traffic-engineering gribi nexthop-group-aft unacknowledged [...]

details_	*boolean*	gRIBI Nexthop Group AFT Detail.
unacknowledgedNhgs	*table(numeric string) = GribiNhgAftVersionType* see 2.1830.1	Map of versioned Nexthop Group AFT entries, keyed by Nexthop Group ID.

2.1830.1 GribiNhgAftVersionType

nhgVersionAft	*table(numeric string) = GribiNhgAftType* see 2.1829.1	Map of Nexthop Group AFT entries, keyed by version ID.

2.1831 show traffic-engineering interfaces [...]

interfaces	*list of TeIntfModel* see 2.1831.1	List of interfaces.

2.1831.1 TeIntfModel

adjSids	*list of AdjSid* see 2.1831.1.1	List of Adjacency SIDs.
adminGroup	*integer*	Administrative groups.
adminGroupNames **	*list of string*	List of Administrative group names. **Since: 4.27.1
delay	*TeDelay* see 2.1831.1.4	Delay parameters.
intf	*interface name*	Interface.
maxResvBw	*TeMaxResvBw* see 2.1831.1.5	Max reservable bandwidth.
metric	*integer*	Traffic engineering metric.
nodeSids	*list of NodeSid* see 2.1831.1.3	List of Node SIDs.
srlg	*list of IntfSrlg* see 2.1831.1.2	List of Shared Risk Link Groups.

2.1831.1.1 AdjSid

index	*integer*	Index of the Segment Identifier.
isIpV6	*boolean*	SID is IPv6.
isValue	*boolean*	Index is a value.

2.1831.1.2 IntfSrlg

gid	*integer*	Shared Risk Link Group identifier.
name	*string*	Shared Risk Link Group name.

2.1831.1.3 NodeSid

flexAlgoName	*string*	Algorithm Name.
index	*integer*	Index of the Segment Identifier.
isIpV6	*boolean*	SID is IPv6.

2.1831.1.4 TeDelay

style	*enum: static | twamp*	Method for determining the delay.
units	*enum: microseconds | milliseconds*	Delay units.
value	*integer*	Interface delay.

2.1831.1.5 TeMaxResvBw

bwUnitType	*enum: percent | gbps | mbps*	Bandwidth units.
value	*integer*	Bandwidth.

2.1832 show traffic-engineering interfaces bandwidth [...]

interfaces	*table(Interface name) = TeInterfaceBwModel* see 2.1832.1	Dictionary of interface bandwidth statistics.
threshold	*integer*	IGP unreserved bandwidth update threshold in percentage.

2.1832.1 TeInterfaceBwModel

advertisedUnreserved-Bandwidths	*table(string) = BandwidthAdByIgp* see 2.1832.1.1	Dictionary of advertised unreserved bandwidth by IGP.
maxBandwidth	*integer*	Interface maximum bandwidth in bps.
maxReservableBandwidth	*integer*	Interface maximum reservable bandwidth in bps.
unreservedBandwidths	*table(numeric string) = integer*	Current unreserved bandwidth for each priority.

2.1832.1.1 BandwidthAdByIgp

bandwidthsInEachLevelOrArea	*table(numeric string) = AdvertisedBwOfPriority-Classes* see 2.1832.1.1.1	Dictionary of advertised unreserved bandwidth in each IGP level or area.
instanceName	*string*	IGP Instance name.

2.1832.1.1.1 AdvertisedBwOfPriorityClasses

bandwidths	*table(numeric string) = integer*	Advertised unreserved bandwidth for each priotiy in bps.

2.1833 show traffic-engineering rsvp

autoBwSupported **	*boolean*	Is auto bandwidth supported. **Since: 4.25.1
localIntf	*interface name*	Local interface.
localIntfConfigured **	*boolean*	Is the local interface configured. **Since: **currentRelease**
localIntfIp	*IP address*	Local interface IP address.
lspCount	*integer*	Number of LSPs.
lspUpCount	*integer*	Number of LSPs that are up.
optimizationInterval **	*integer*	Optimization interval value in seconds. **Since: 4.25.1

[splitBwTunnelCount] **	*integer*	Number of tunnels with split bandwidth configured. **Since: 4.26.2
[subTunnelCount] **	*integer*	Number of sub tunnels. **Since: 4.26.2
[subTunnelSecondaryInUse-Count] **	*integer*	Number of sub tunnels with secondary path in use. **Since: 4.26.2
[subTunnelUpCount] **	*integer*	Number of sub tunnels that are up. **Since: 4.26.2
tunnelCount	*integer*	Number of tunnels.
tunnelSecondaryEnabledCount	*integer*	Number of tunnels with secondary path configured.
tunnelSecondaryInUseCount	*integer*	Number of tunnels with secondary path in use.
tunnelUpCount	*integer*	Number of tunnels that are up.

2.1834 show traffic-engineering rsvp tunnel [...]

tunnels	*table(string) = RsvpLerTunnel* see 2.1834.1	Table of tunnel information keyed by tunnel name.

2.1834.1 RsvpLerTunnel

activePathType	*enum: secondary | primary | unknown | split*	Active path type.
[aliasTeps] **	*list of IP address*	Tunnel endpoint aliases. **Since: 4.26.0
autoBandwidth	*boolean*	Bandwidth is automatically determined.
[autoBandwidthParams]	*RsvpLerAutoBandwidthParams* see 2.1834.1.1	Parameters for tuning automagic bandwidth determination.
[color] **	*integer*	Tunnel color. **Since: 4.26.0
[configSource] **	*enum: frrBypass | cli | unknown*	Tunnel configuration source. **Since: **currentRelease**
currentBandwidth	*integer*	Currently reserved bandwidth in bps.
destination	*IP address*	Destination address.
[holdPriority]	*integer*	Hold priority.
[igpShortcut] **	*boolean*	Tunnel eligible to be used for IGP shortcuts. **Since: 4.25.1
[ldpTunneling] **	*boolean*	Tunnel eligible to be used for LDP tunneling. **Since: 4.25.1
[ldpTunnelingUcmp] **	*boolean*	UCMP enabled on tunnel for LDP tunneling. **Since: 4.26.2
lspCount	*integer*	Number of LSPs in this tunnel.
[lspHistory]	*table(numeric string) = RsvpLerHistory* see 2.1834.1.2	LSP history events keyed by LSP ID.
[lsps]	*list of RsvpLerLsp* see 2.1834.1.3	LSPs part of this tunnel.
mbb	*boolean*	Tunnel is getting re-established with make-before-break.
metric **	*integer*	Tunnel metric. **Since: 4.26.2
metricMode **	*enum: dynamic | static*	Tunnel metric mode. **Since: 4.30.0
[mtuSignalingEnabled]	*boolean*	MTU Signaling is enabled.
[optimizationInterval] **	*integer*	Optimization interval value in seconds. **Since: 4.25.1
[primaryPath]	*RsvpLerTunnelPath* see 2.1834.1.5.1	Primary path information.
[profile] **	*string*	Tunnel profile name. **Since: 4.29.2
requestedBandwidth	*integer*	Requested bandwidth in bps.

[secondaryPaths]	*list of RsvpLerTunnelPath* see 2.1834.1.5.1	Secondary paths information.
[sessionId] **	*integer*	Session ID associated with this tunnel. **Since: 4.25.1
[setupPriority]	*integer*	Setup priority.
[signalBandwidth] **	*boolean*	Bandwidth is signaled. **Since: 4.26.2
source	*IP address*	Destination address.
splitTunnel **	*boolean*	Tunnel has split-tunnel enabled. **Since: 4.26.2
[splitTunnelParams] **	*RsvpLerSplitTunnelParams* see 2.1834.1.4	Parameters for split tunnel. **Since: 4.26.2
state	*enum: down | up | unknown*	Tunnel state.
[subTunnels] **	*table(numeric string) = RsvpLerSubTunnel* see 2.1834.1.5	Sub tunnels part of this tunnel keyed by sub tunnel ID. **Since: 4.26.2
[tunnelHistory]	*RsvpLerHistory* see 2.1834.1.2	Tunnel history events.
[tunnelIndex]	*integer*	Index of this tunnel in the tunnel FIB.

2.1834.1.1 RsvpLerAutoBandwidthParams

adjustmentPeriod	*floating point number*	Bandwidth adjustment period for auto bandwidth.
lastSampledBandwidth **	*integer*	Last sampled bandwidth for auto bandwidth in bps. **Since: 4.25.1
maxBandwidth	*integer*	Maximum bandwidth for auto bandwidth in bps.
minBandwidth	*integer*	Minimum bandwidth for auto bandwidth in bps.

2.1834.1.2 RsvpLerHistory

history	*list of RsvpLerHistoryEntry* see 2.1834.1.2.1	Recent history events.

2.1834.1.2.1 RsvpLerHistoryEntry

event	*string*	History event.
timestamp	*floating point number*	Timestamp of the event.

2.1834.1.3 RsvpLerLsp

bandwidth	*integer*	LSP requested bandwidth in bps.
[cspfPathKeyId]	*integer*	CSPF Path ID.
hops	*list of IP address*	Path hops.
inUse	*boolean*	LSP is in use.
lspErrors	*list of RsvpLerError* see 2.1834.1.3.1	Errors on this LSP.
lspId	*integer*	LSP ID.

pathSpecName	*string*	Path specification name.
pathSpecType	*enum: explicit* | *dynamic* | *unknown*	Path specification type.
pathType	*enum: secondary* | *primary* | *unknown*	Path type.
state	*enum: unknown* | *establishing* | *down* | *unverified* | *up*	LSP state.

2.1834.1.3.1 RsvpLerError

code	*integer*	RSVP error code.
node	*IP address*	Router on which the error occurred.
value	*integer*	RSVP error value.

2.1834.1.4 RsvpLerSplitTunnelParams

maxBandwidth **	*integer*	Maximum bandwidth for each sub-tunnel in bps. **Since: 4.27.0
minBandwidth **	*integer*	Minimum bandwidth for each sub-tunnel in bps. **Since: 4.27.0
quantum	*integer*	Quantum bandwidth for split-tunnel in bps.
reductionDelay **	*integer*	Sub-tunnel reduction delay interval value. **Since: 4.27.1
subTunnelLimit	*integer*	Limit on the number of sub-tunnels for split-tunnel.

2.1834.1.5 RsvpLerSubTunnel

activePathType	*enum: secondary* | *primary* | *unknown*	Active path type.
currentBandwidth	*integer*	Sub tunnel bandwidth in bps.
[lastSampledBandwidth] **	*integer*	Last sampled bandwidth of this sub tunnel in bps. **Since: 4.27.1
lspCount	*integer*	Number of LSPs in this sub tunnel.
[lspHistory]	*table(numeric string) = RsvpLerHistory* see 2.1834.1.2	LSP history events keyed by LSP ID.
[lsps]	*list of RsvpLerLsp* see 2.1834.1.3	LSPs part of this sub tunnel.
mbb	*boolean*	Sub tunnel is getting re-established with make-before-break.
[primaryPath]	*RsvpLerTunnelPath* see 2.1834.1.5.1	Primary path information.
[secondaryPaths]	*list of RsvpLerTunnelPath* see 2.1834.1.5.1	Secondary paths information.
[sessionId]	*integer*	Session ID associated with this sub tunnel.
state	*enum: down* | *up* | *unknown*	Sub tunnel state.

[subTunnelHistory]	*RsvpLerHistory* see 2.1834.1.2	Sub tunnel history events.
[tunnelIndex]	*integer*	Index of this sub tunnel in the tunnel FIB.

2.1834.1.5.1 RsvpLerTunnelPath

[cspfPathKeyId]	*integer*	CSPF Path ID.
hops	*list of IP address*	Path hops.
inUse	*boolean*	Path is in use.
[intfMtu]	*integer*	Interface MTU.
mbb	*boolean*	Path is being re-established with make-before-break.
pathErrors	*list of RsvpLerError* see 2.1834.1.3.1	Errors on this path.
[pathMtu]	*integer*	Path MTU.
[preSignaled]	*boolean*	Path is pre-signaled.
specName	*string*	Path specification name.
specType	*enum: explicit | dynamic | unknown*	Path specification type.
state	*enum: unknown | establishing | down | coldStandby | up*	Path state.

2.1835 show traffic-engineering segment-routing cbf fec [...]

vrfs	*table(string) = SrTePolicyCbfFecModel* see 2.1835.1	Per VRF Segment Routing Traffic Engineering CBF overrideentries.

2.1835.1 SrTePolicyCbfFecModel

overrides	*list of SrTePolicyCbfFecOverride* see 2.1835.1.1	Policy FEC override entry.

2.1835.1.1 SrTePolicyCbfFecOverride

defaultFecId	*integer*	Default FEC ID.
dscp	*integer*	DSCP value.
ipVersion	*enum: ipv4 | ipv6*	IP version.
overrideFecId	*integer*	Overriding SR-TE Policy FEC ID.

2.1836 show traffic-engineering segment-routing policy [...]

vrfs	*table(string) = SrTePolicyCandidatesModel* see 2.1836.1	Per VRF Segment Routing Traffic Engineering Policies.

2.1836.1 SrTePolicyCandidatesModel

enlpGlobal **	*enum: localPolicy | ipv4 | ipv6 | ipv4AndIpv6 | none*	Global explicit null label policy. **Since: 4.25.2
policies	*list of SrTePolicyCandidates* see 2.1836.1.1	List of Segment Routing Traffic Engineering policies and its candidates.

2.1836.1.1 SrTePolicyCandidates

color	*integer*	Segment Routing Traffic Engineering Policy Color.
[description] **	*string*	Policy Description. **Since: 4.24.1
endpoint	*IP address*	Segment Routing Traffic Engineering Policy End point.
[name] **	*string*	Policy Name. **Since: 4.24.1
pathGroups	*list of SrTePolicyPathGroup* see 2.1836.1.1.1	Candidate paths for the Segment Routing Traffic Engineering policy.
[txBytes] **	*integer*	Aggregate number of transmitted bytes for the policy. **Since: 4.22.0
[txPackets] **	*integer*	Aggregate number of transmitted packets for the policy. **Since: 4.22.0

2.1836.1.1.1 SrTePolicyPathGroup

[activeTime]	*integer*	UTC seconds since epoch when the candidate path was selected as active.
[bindingSid]	*SrTePolicyBindingSid* see 2.1836.1.1.1.1	Binding Segment Identifier of the candidate path.
[discriminator]	*integer*	Protocol specific discriminator of the candidate path.
enlp **	*enum: localPolicy | ipv4 | ipv6 | ipv4AndIpv6 | none*	Explicit null label policy of the candidate path. **Since: 4.23.1
inheritedGlobalEnlp **	*boolean*	Flag to indicate inheritance of Global explicit null label policy. **Since: 4.25.2
[lastChangeTime]	*integer*	UTC seconds since epoch when the candidate path was last updated.
[metric] **	*integer*	Metric of the candidate path. **Since: 4.28.1
[metricType] **	*enum: dynamicUnresolved | static | dynamicResolved*	Static or dynamic metric. **Since: 4.28.1

[notActiveReason]	*enum: higherPreferenceAvailable | bindingSidOutOfSrlbRange | betterProtocolAvailable | higherDiscriminator-Available | segmentListsUnresolved | segmentListsMissing | bindingSidInUseByOther-Application | bindingSidNotConfigured | bindingSidConflict | lowerOriginAvailable | samePriorityActive*	Reason why this candidate path is not active.
originator	*SrTePolicyOriginator* see 2.1836.1.1.1.2	Originator of the candidate path.
preference	*integer*	Preference of the candidate path.
protocol	*enum: static | bgp*	Protocol source of the candidate path.
segmentLists	*list of SrTePolicyPath-GroupSegmentList* see 2.1836.1.1.1.3	Segment lists that make up the candidate path.
state	*enum: valid | invalid | active*	State of the candidate path.

2.1836.1.1.1.1 SrTePolicyBindingSid

bindingSidType	*enum: mpls | ipv6*	Binding Segment Identifier Type.
[mplsLabelSid]	*integer*	Binding Segment Identifier in the form of an MPLS label.

2.1836.1.1.1.2 SrTePolicyOriginator

asn	*string*	Autonomous System Number.
nodeAddress	*IP address*	Node address of the originator.

2.1836.1.1.1.3 SrTePolicyPathGroupSegmentList

[backupVias] **	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the backup path of the segment list, if the segment list is valid. **Since: 4.23.2
[counterInUse] **	*boolean*	Counter is in use for the segment list. **Since: 4.30.1
[invalidReason]	*enum: sbfdDown | segmentListMissing | resolvedLabelsExceed-PlatformMsd | unresolvedTopLabel | noResolvedLabels*	Reason why this segment list is invalid.
[protectionState] **	*enum: slProtected | protectionUnknown | unprotectedTilfaBackup-ExceedMSD*	Segment list protection status and the reason why unprotected. **Since: 4.30.1

[sbfdReturnPathLabelStack] **	*list of SrTeSegment* see 2.1836.1.1.1.4	Segments that make up the SBFD return path. **Since: 4.29.2
[sbfdState] **	*enum: down | adminDown | up | init*	State of SBFD session. **Since: 4.23.2
[segmentListId]	*integer*	Internal Id of the segment list for valid candidate paths.
[segmentListValid]	*boolean*	Validity of the segment list for valid candidate paths.
segments	*list of SrTeSegment* see 2.1836.1.1.1.4	Segments that make up the segment list.
[txBytes]	*integer*	Total number of transmitted bytes.
[txPackets]	*integer*	Total number of transmitted packets.
[vias]	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the primary path of the segment list, if the segment list is valid.
weight	*integer*	Weight of the segment list.

2.1836.1.1.1.4 SrTeSegment

[mplsLabelSid]	*integer*	Segment Identifier in the form of an MPLS label.

2.1837 show traffic-engineering segment-routing policy active [...]

vrfs	*table(string) = SrTePolicyActiveModel* see 2.1837.1	Per VRF Active Segment Routing Traffic Engineering Policies.

2.1837.1 SrTePolicyActiveModel

enlpGlobal **	*enum: localPolicy | ipv4 | ipv6 | ipv4AndIpv6 | none*	Global explicit null label policy. **Since: 4.25.2
policies	*list of SrTePolicyActive* see 2.1837.1.1	List of Segment Routing Traffic Engineering policies installed in the forwarding plane.

2.1837.1.1 SrTePolicyActive

color	*integer*	Segment Routing Traffic Engineering Policy Color.
[description] **	*string*	Policy Description. **Since: 4.24.1
endpoint	*IP address*	Segment Routing Traffic Engineering Policy End point.
[name] **	*string*	Policy Name. **Since: 4.24.1
pathGroup	*SrTePolicyPathGroup* see 2.1836.1.1.1	Selected candidate path for the Segment Routing Traffic Engineering policy.
[txBytes] **	*integer*	Aggregate number of transmitted bytes for the policy. **Since: 4.22.0

[txPackets] **	*integer*	Aggregate number of transmitted packets for the policy. **Since: 4.22.0

2.1838 show traffic-engineering segment-routing policy fec [...]

[fecs]	*table(numeric string) = SrTePolicyFecInfo* see 2.1838.1	A mapping from policy FEC ID to policy information.

2.1838.1 SrTePolicyFecInfo

[policy]	*SrTePolicyBase* see 2.1838.1.1	Segment routing traffic engineering policy information.
[vias]	*list of SrTePolicyFecVia* see 2.1838.1.2	List storing the policy FEC vias.

2.1838.1.1 SrTePolicyBase

Could also be an instance of:
SrTePolicyActive 2.1837.1.1
SrTePolicyCandidates 2.1836.1.1

color	*integer*	Segment Routing Traffic Engineering Policy Color.
[description] **	*string*	Policy Description. **Since: 4.24.1
endpoint	*IP address*	Segment Routing Traffic Engineering Policy End point.
[name] **	*string*	Policy Name. **Since: 4.24.1
[txBytes] **	*integer*	Aggregate number of transmitted bytes for the policy. **Since: 4.22.0
[txPackets] **	*integer*	Aggregate number of transmitted packets for the policy. **Since: 4.22.0

2.1838.1.2 SrTePolicyFecVia

segmentListId	*integer*	Segment list identifier.
weight	*integer*	Segment list weight.

2.1839 show traffic-engineering segment-routing policy segment-list [...]

vrfs	*table(string) = SrTePolicySegmentLists* see 2.1839.1	A mapping from VRF to the segment lists it contains.

2.1839.1 SrTePolicySegmentLists

segmentLists	*table(numeric string) = SrTePolicySegmentList* see 2.1839.1.1	A mapping from segment list ID to the segment list.

2.1839.1.1 SrTePolicySegmentList

[backupVias] **	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of nexthops for the backup path of the segment list. **Since: 4.23.2
segments	*list of SrTeSegment* see 2.1836.1.1.1.4	List of segments composing the segment list.
valid	*boolean*	Validity of the segment list.
[vias]	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the primary path of the segment list.

2.1840 show traffic-engineering segment-routing policy segment-list counters [...]

vrfs	*table(string) = SrTePolicySegmentListsCounters* see 2.1840.1	A mapping from VRF to the segment lists it contains.

2.1840.1 SrTePolicySegmentListsCounters

segmentLists	*table(numeric string) = SrTePolicySegmentListCounters* see 2.1840.1.1	A mapping from segment list ID to the segment list counters.

2.1840.1.1 SrTePolicySegmentListCounters

[counterInUse] **	*boolean*	Counter is in use for the segment list. **Since: 4.30.1
[txBytes]	*integer*	Total number of transmitted bytes.
[txPackets]	*integer*	Total number of transmitted packets.
[vias]	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the segment list.

2.1841 show traffic-engineering segment-routing policy summary

revision 2

vrfs	*table(string) = SrTePolicySummaryModel* see 2.1841.1	Per VRF Segment Routing Traffic Engineering Policy summary.

2.1841.1 SrTePolicySummaryModel

bindingSidConflictInactive-PoliciesCount	*integer*	Number of inactive policies due to Binding SID conflict with other policies.
bindingSidConflictInvalid-PathGroupsCount	*integer*	Number of invalid candidate paths due to Binding SID conflict with other applications.
fecOptimizationsSupported	*boolean*	Indicates whether this device supports FEC optimizations or not.
pathgroupStatistics	*SrTePolicyPathGroup-StatisticsModel* see 2.1841.1.1	Segment Routing Traffic Engineering Policy path group related Statistics.
perEndpointColorStatistics	*SrTePolicyStatisticsEntry-Model* see 2.1841.1.3	Per endpoint and color Segment Routing Traffic Engineering Policy statistics.
policyStatistics	*SrTePolicyStatisticsModel* see 2.1841.1.4	Per endpoint and color and source Segment Routing Traffic Engineering Policy statistics.
segmentListStatistics	*SrTePolicySegmentList-StatisticsModel* see 2.1841.1.2	Segment Routing Traffic Engineering Policy segment list related statistics.

2.1841.1.1 SrTePolicyPathGroupStatisticsModel

sources	*table(string) = SrTePolicyPathGroup-StatisticsEntryModel* see 2.1841.1.1.1	Per source Segment Routing Traffic Engineering Policy path group statistics.
total	*SrTePolicyPathGroup-StatisticsEntryModel* see 2.1841.1.1.1	Total path group statistics for all Segment Routing Traffic Engineering Policy sources.

2.1841.1.1.1 SrTePolicyPathGroupStatisticsEntryModel

activePathGroups	*integer*	Number of candidate paths in active state.
invalidPathGroups	*integer*	Number of candidate paths in invalid state.
totalPathGroups	*integer*	All candidate paths in any state.
validPathGroups	*integer*	Number of candidate paths that are valid but not active.

2.1841.1.2 SrTePolicySegmentListStatisticsModel

invalidSegmentLists	*integer*	Number of segment lists in invalid state.
validSegmentLists	*integer*	Number of segment lists in valid state.

validSegmentListsWithBackup **	*integer*	Number of valid segment lists with backup. **Since: 4.23.2

2.1841.1.3 SrTePolicyStatisticsEntryModel

activePolicies	*integer*	Number of policies in active state.
invalidPolicies	*integer*	Number of policies in invalid state.
totalPolicies	*integer*	All policies in any state.
validPolicies	*integer*	Number of policies that are valid but not active.

2.1841.1.4 SrTePolicyStatisticsModel

sources	*table(string) = SrTePolicyStatisticsEntryModel* see 2.1841.1.3	Per source Segment Routing Traffic Engineering Policy statistics.
total	*SrTePolicyStatisticsEntryModel* see 2.1841.1.3	Total Segment Routing Traffic Engineering policies with unique Endpoint, Color values irresepctive of source.

2.1842 show traffic-engineering segment-routing replication-segment [...]

vrfs	*table(string) = SrP2mpCandidatesModel* see 2.1842.1	A mapping of VRF name to its replication segments.

2.1842.1 SrP2mpCandidatesModel

replicationSegments	*table(numeric string) = SrP2mpCandidates* see 2.1842.1.1	Replication segments, keyed by their replication ID.

2.1842.1.1 SrP2mpCandidates

nodeAddress	*IP address*	Address of replication node.
pathInstance	*integer*	Path Instance number.
[replicationSid]	*SrP2mpReplicationSid* see 2.1842.1.1.2	Mpls label to identify replication segment.
root	*IP address*	Node address of root.
segmentLists	*list of SrP2mpRepSegSegmentList* see 2.1842.1.1.1	Replication segments information.
source	*enum: pcep | static*	Protocol source of replication segment.
state	*enum: valid | invalid | active*	State of the replication segment.
treeId	*integer*	Tree identifier.

2.1842.1.1.1 SrP2mpRepSegSegmentList

backupVias	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the backup path of the segment list, if the segment list is valid.
[invalidReason]	*enum: sbfdDown | segmentListMissing | resolvedLabelsExceed-PlatformMsd | unresolvedTopLabel | noResolvedLabels*	Reason why this segment list is invalid.
[segmentListId]	*integer*	Internal ID of the segment list for valid candidate paths.
[segmentListValid]	*boolean*	Validity of the segment list for valid candidate paths.
segments	*list of SrP2mpSegment* see 2.1842.1.1.1.1	Segments that make up the segment list.
vias	*list of SrTeSegmentListVia* see 2.1826.1.1.1.2	List of next hops for the primary path of the segment list, if the segment list is valid.

2.1842.1.1.1.1 SrP2mpSegment

[mplsLabelSid]	*integer*	Segment identifier in the form of an MPLS label.

[nexthopAddr]	*IP address*	Segment identifier in the form of next hop IP address.

2.1842.1.1.2 SrP2mpReplicationSid

[mplsLabelSid]	*integer*	Replication Segment Identifier in the form of an MPLS label.
replicationSidType	*enum: mpls | ipv6*	Replicaiton Segment Identifier Type.

2.1843 show traffic-engineering segment-routing replication-segment summary

vrfs	*table(string) = ReplicationSegmentSummaryModel* see 2.1843.1	Per VRF Segment Routing Replication Segments.

2.1843.1 ReplicationSegmentSummaryModel

nonSharedStatistics	*ReplicationSegmentRoot* see 2.1843.1.1	Segment Routing Traffic Engineering not shared Replication Segment related statistics.
protocolStats	*ReplicationSegmentStatisticsModel* see 2.1843.1.3	Per source protocol Routing Traffic Engineering Replication segment statistics.
replicationSegmentsRsidConflictCount	*integer*	Number of replication segment with RSID conflict.
sharedStatistics	*ReplicationSegmentSharedStatModel* see 2.1843.1.2	Segment Routing Traffic Engineering shared Replication Segment related statistics.

2.1843.1.1 ReplicationSegmentRoot

roots	*table(IPv4 or IPv6 address) = ReplicationSegmentTree* see 2.1843.1.1.1	Replication segments, keyed by their root address.

2.1843.1.1.1 ReplicationSegmentTree

trees	*table(numeric string) = ReplicationSegmentPath* see 2.1843.1.1.1.1	Replication segments, keyed by their tree ID.

2.1843.1.1.1.1 ReplicationSegmentPath

paths	*table(numeric string) = ReplicationSegmentStateModel* see 2.1843.1.1.1.2	Replication segments, keyed by their path instance number.

2.1843.1.1.1.2 ReplicationSegmentStateModel

active	*integer*	Number of replication segments in active state.
invalid	*integer*	Number of replication segments in invalid state.
valid	*integer*	Number of replication segments in valid state.

2.1843.1.2 ReplicationSegmentSharedStatModel

`replicationIds`	*table(numeric string) = ReplicationSegmentStateModel* see 2.1843.1.1.1.2	Replication segments, keyed by their replication id.

2.1843.1.3 ReplicationSegmentStatisticsModel

`sources`	*table(string) = ReplicationSegmentStateModel* see 2.1843.1.1.1.2	Replication segments, keyed by their source.
`totalStat`	*ReplicationSegmentStateModel* see 2.1843.1.1.1.2	Total replication segment statistics for all sources.

2.1844 show traffic-policy (cpu | vlan | interface) [...]

revision 2

[trafficPolicies]	*table(string) = TrafficPolicy* see 2.1844.1	Traffic policy indexed by policy name.

2.1844.1 TrafficPolicy

[configuredIntfs]	*list of Interface name*	(Deprecated - refer to the "configuredIntfs" attribute in the "input" InterfaceCommon sub-model) List of interfaces the policy is configured on input.
[configuredVrfs]	*list of string*	List of configured VRFs.
[input] **	*InterfaceCommon* see 2.1844.1.1	Interface information for input policy. **Since: 4.26.0
[ipv4AppliedIntfs]	*list of Interface name*	(Deprecated - refer to the "ipv4AppliedIntfs" attribute in the "input" InterfaceCommon sub-model) Applied interfaces for IPv4 input traffic.
[ipv4AppliedVrfs]	*list of string*	Applied VRFs for IPv4 traffic (enforced in hardware).
[ipv4AppliedVrfsSoftware] **	*list of string*	Applied VRFs for IPv4 traffic (enforced in software). **Since: 4.27.1
[ipv6AppliedIntfs]	*list of Interface name*	(Deprecated - refer to the "ipv6AppliedIntfs" attribute in the "input" InterfaceCommon sub-model) Applied interfaces for IPv6 input traffic.
[ipv6AppliedVrfs]	*list of string*	Applied VRFs for IPv6 traffic (enforced in hardware).
[ipv6AppliedVrfsSoftware] **	*list of string*	Applied VRFs for IPv6 traffic (enforced in software). **Since: 4.27.1
[namedCounters]	*list of string*	Defined named counters for a traffic-policy.
[output] **	*InterfaceCommon* see 2.1844.1.1	Interface information for output policy. **Since: 4.26.0
rules	*list of Rule* see 2.1038.2.1	Detailed information of match rules.
[vacl] **	*VAclCommon* see 2.1844.1.2	Vlan information for VAcl policy. **Since: 4.27.2

2.1844.1.1 InterfaceCommon

[configuredIntfs]	*list of Interface name*	List of configured interfaces.
[ipv4AppliedIntfs]	*list of Interface name*	Applied interfaces for IPv4 traffic.
[ipv6AppliedIntfs]	*list of Interface name*	Applied interfaces for IPv6 traffic.
[macAppliedIntfs] **	*list of Interface name*	Applied interfaces for MAC traffic. **Since: 4.29.2

2.1844.1.2 VAclCommon

[configuredVlans]	*list of string*	List of configured vlans.
[ipv4AppliedVlans]	*list of string*	Applied vlans for IPv4 traffic.
[ipv6AppliedVlans]	*list of string*	Applied vlans for IPv6 traffic.

2.1845 show traffic-policy (cpu | vlan | interface) counters [...]

[egressCounterGranularity] **	*enum: per-interface | per-policy*	Enabled counter granularity for traffic-policies applied on output of interfaces. **Since: 4.26.0
[egressTrafficPolicies] **	*table(string) = TrafficPolicyCounters* see 2.1845.1	Traffic policies applied on output of interfaces indexed by policy name. **Since: 4.26.0
[ingressCounterGranularity] **	*enum: per-interface | per-policy*	Enabled counter granularity for traffic-policies applied on input of interfaces. **Since: 4.25.0
[trafficPolicies]	*table(string) = TrafficPolicyCounters* see 2.1845.1	Traffic policy indexed by policy name. Contains either policies applied on input of interfaces or CPU policies.

2.1845.1 TrafficPolicyCounters

[namedCounters]	*table(string) = CounterInfo* see 2.1845.1.1	Mapping of named counter to count data.
[unnamedCounters]	*table(string) = CounterInfo* see 2.1845.1.1	Mapping of rule name to count data.

2.1845.1.1 CounterInfo

[interfaces]	*table(string) = InterfaceInfo* see 2.1845.1.1.1	Count data per interface.
[totals]	*TotalInfo* see 2.1845.1.1.2	Total hit counts.

2.1845.1.1.1 InterfaceInfo

[byteCount]	*integer*	Number of bytes matched.
[packetCount]	*integer*	Number of packets matched.

2.1845.1.1.2 TotalInfo

[byteTotal]	*integer*	Total byte count.
[packetTotal]	*integer*	Total packet count.

2.1846 show traffic-policy (cpu | vlan | interface) errors [...]

revision 2

[trafficPolicies]	*table(string) = TrafficPolicyErrors* see 2.1846.1	Traffic policy indexed by policy name.

2.1846.1 TrafficPolicyErrors

[input] **	*TrafficPolicyIntfErrorInfo* see 2.1846.1.1	Installation errors for policies applied on input of interfaces. **Since: 4.26.0
[ipv4FailedIntfs]	*list of Interface name*	(Deprecated - refer to "ipv4FailedIntfs" in the "input" TrafficPolicyIntfErrorInfo sub-model) Failed interfaces for IPv4 traffic.
[ipv4FailedVrfs]	*list of string*	Failed VRFs for IPv4 traffic (enforced in hardware).
[ipv4FailedVrfsSoftware] **	*list of string*	Failed VRFs for IPv4 traffic (enforced in software). **Since: 4.27.1
[ipv4InstallationErrors]	*list of string*	List of IPv4 installation errors. For traffic-policies applied on interfaces, refer to the "ipv4InstallationErrors" in the input/output TrafficPolicyIntfErrorInfo sub-models.
[ipv4InstallationErrors-Software] **	*list of string*	List of IPv4 installation errors. For traffic-policies applied on interfaces, refer to the "ipv4InstallationErrors" in the input/output TrafficPolicyIntfErrorInfo sub-models. **Since: 4.27.1
[ipv6FailedIntfs]	*list of Interface name*	(Deprecated - refer to "ipv6FailedIntfs" in the "input" TrafficPolicyIntfErrorInfo sub-model) Failed interfaces for IPv6 traffic.
[ipv6FailedVrfs]	*list of string*	Failed VRFs for IPv6 traffic (enforced in hardware).
[ipv6FailedVrfsSoftware] **	*list of string*	Failed VRFs for IPv6 traffic (enforced in software). **Since: 4.27.1
[ipv6InstallationErrors]	*list of string*	List of IPv6 installation errors. For traffic-policies applied on interfaces, refer to the "ipv6InstallationErrors" in the input/output TrafficPolicyIntfErrorInfo sub-models.
[ipv6InstallationErrors-Software] **	*list of string*	List of IPv6 installation errors. For traffic-policies applied on interfaces, refer to the "ipv6InstallationErrors" in the input/output TrafficPolicyIntfErrorInfo sub-models. **Since: 4.27.1
[macInstallationErrors] **	*list of string*	List of MAC installation errors. For traffic-policies applied on interfaces, refer to the "macInstallationErros" in the input/output TrafficPolicyIntfErrorInfo sub-models. **Since: 4.29.2

[output] **	*TrafficPolicyIntfErrorInfo* see 2.1846.1.1	Installation errors for policies applied on output of interfaces. **Since: 4.26.0
[vacl] **	*TrafficPolicyVAclErrorInfo* see 2.1846.1.2	Installation errors for policies applied on vlan. **Since: 4.27.2

2.1846.1.1 TrafficPolicyIntfErrorInfo

[ipv4FailedIntfs]	*list of Interface name*	Failed interfaces for IPv4 traffic.
[ipv4InstallationErrors]	*list of string*	List of IPv4 installation errors.
[ipv6FailedIntfs]	*list of Interface name*	Failed interfaces for IPv6 traffic.
[ipv6InstallationErrors]	*list of string*	List of IPv6 installation errors.
[macFailedIntfs] **	*list of Interface name*	Failed interfaces for MAC traffic. **Since: 4.29.2
[macInstallationErrors] **	*list of string*	List of MAC installation errors. **Since: 4.29.2

2.1846.1.2 TrafficPolicyVAclErrorInfo

[ipv4FailedVlans]	*list of string*	Failed vlans for IPv4 traffic.
[ipv4InstallationErrors]	*list of string*	List of IPv4 installation errors.
[ipv6FailedVlans]	*list of string*	Failed vlans for IPv6 traffic.
[ipv6InstallationErrors]	*list of string*	List of IPv6 installation errors.

2.1847 show traffic-policy (cpu | vlan | interface) summary [...]

[trafficPolicies]	*table(string) = TrafficPolicySummary* see 2.1847.1	Traffic policy indexed by policy name.

2.1847.1 TrafficPolicySummary

[configuredIntfs]	*list of Interface name*	(Deprecated - refer to the "configuredIntfs" attribute in the "input" InterfaceCommon sub-model) List of interfaces the policy is configured on input.
[configuredVrfs]	*list of string*	List of configured VRFs.
[input] **	*InterfaceCommon* see 2.1844.1.1	Interface information for input policy. **Since: 4.26.0
[ipv4AppliedIntfs] **	*list of Interface name*	(Deprecated - refer to the "ipv4AppliedIntfs" attribute in the "input" InterfaceCommon sub-model) Applied interfaces for IPv4 input traffic. **Since: 4.23.0
[ipv4AppliedVrfs]	*list of string*	Applied VRFs for IPv4 traffic (enforced in hardware).
[ipv4AppliedVrfsSoftware] **	*list of string*	Applied VRFs for IPv4 traffic (enforced in software). **Since: 4.27.1
[ipv6AppliedIntfs] **	*list of Interface name*	(Deprecated - refer to the "ipv6AppliedIntfs" attribute in the "input" InterfaceCommon sub-model) Applied interfaces for IPv6 input traffic. **Since: 4.23.0
[ipv6AppliedVrfs]	*list of string*	Applied VRFs for IPv6 traffic (enforced in hardware).
[ipv6AppliedVrfsSoftware] **	*list of string*	Applied VRFs for IPv6 traffic (enforced in software). **Since: 4.27.1
[output] **	*InterfaceCommon* see 2.1844.1.1	Interface information for output policy. **Since: 4.26.0
rules	*list of RuleSummary* see 2.1847.1.1	Summary of match rules configured.
[vacl] **	*VAclCommon* see 2.1844.1.2	Vlan information for VAcl policy. **Since: 4.27.2

2.1847.1.1 RuleSummary

Could also be an instance of:
Rule 2.1038.2.1

matchOption	*string*	Match on packet type.
ruleString	*string*	Match rule name.

2.1848 show traffic-policy field-set integer [...]

[integerFieldSets]	*table(string) = TrafficPolicyIntegerFieldSet* see 2.1848.1	Integer field sets by name.

2.1848.1 TrafficPolicyIntegerFieldSet

[integers]	*list of NumericalRange* see 2.1038.2.1.2.2	Integer ranges.
[policies] **	*PolicyUsage* see 2.1848.1.1	Policies applied with the field-set. **Since: 4.29.1
source	*enum: url | other | static | bgp*	Contents source type.
[urlInfo]	*UrlInfo* see 2.1848.1.2	URL information.

2.1848.1.1 PolicyUsage

[policyNames]	*list of string*	policies applied with the field-set.

2.1848.1.2 UrlInfo

[urlPath]	*string*	Path to URL file.
[vrf]	*string*	Name of VRF for URL fetch.

2.1849 show traffic-policy field-set ipv4 prefix [...]

[prefixFieldSets]	*table(string) = IpPrefixFieldSet* see 2.1849.1	Prefix field sets by name.

2.1849.1 IpPrefixFieldSet

[policies] **	*PolicyUsage* see 2.1848.1.1	Policies applied with the field-set. **Since: 4.29.1
[prefixes]	*list of IpPrefixWithExceptions* see 2.1849.1.1	Allowed prefixes with associated exception prefixes.
source	*enum: url | other | static | bgp*	Contents source type.
[urlInfo] **	*UrlInfo* see 2.1848.1.2	URL information. **Since: 4.27.1

2.1849.1.1 IpPrefixWithExceptions

[exceptPrefixes]	*list of IP address prefix*	Exception prefixes.
[prefix]	*IP address prefix*	Allowed prefix.

2.1850 show traffic-policy field-set ipv6 prefix [...]

see 2.1849

2.1851 show traffic-policy field-set l4-port [...]

[l4PortFieldSets]	*table(string) = TrafficPolicyL4PortFieldSet* see 2.1851.1	Layer 4 port field sets by name.

2.1851.1 TrafficPolicyL4PortFieldSet

[policies] **	*PolicyUsage* see 2.1848.1.1	Policies applied with the field-set. **Since: 4.29.1
[ports]	*list of NumericalRange* see 2.1038.2.1.2.2	Port ranges.
source	*enum: url | other | static | bgp*	Contents source type.
[urlInfo] **	*UrlInfo* see 2.1848.1.2	URL information. **Since: 4.27.1

2.1852 show traffic-policy field-set mac [...]

[macFieldSets]	*table(string) = TrafficPolicyMacFieldSet* see 2.1852.1	MAC address field sets by name.

2.1852.1 TrafficPolicyMacFieldSet

[macAddrs]	*list of MacAddressInfo* see 2.1852.1.1	MAC addresses with optional mask.
[policies]	*PolicyUsage* see 2.1848.1.1	Policies applied with the field-set.
source	*enum: url | other | static | bgp*	Contents source type.
[urlInfo]	*UrlInfo* see 2.1848.1.2	URL information.

2.1852.1.1 MacAddressInfo

macAddr	*MAC address*	MAC address.

2.1853 show traffic-policy field-set vlan [...]

[vlanFieldSets]	*table(string) = TrafficPolicyVlanFieldSet* see 2.1853.1	VLAN field sets by name.

2.1853.1 TrafficPolicyVlanFieldSet

[policies] **	*PolicyUsage* see 2.1848.1.1	Policies applied with the field-set. **Since: 4.29.1
source	*enum: url | other | static | bgp*	Contents source type.
[urlInfo] **	*UrlInfo* see 2.1848.1.2	URL information. **Since: 4.27.1
[vlans]	*list of NumericalRange* see 2.1038.2.1.2.2	VLAN ranges.

2.1854 show traffic-policy protocol neighbors bgp

policies	*table(string) = ProtocolNeighbor* see 2.1854.1	Neighbors per policy.

2.1854.1 ProtocolNeighbor

vrfs	*table(string) = VrfRules* see 2.1854.1.1	Neighbors per VRF.

2.1854.1.1 VrfRules

rules	*list of NeighborRule* see 2.1854.1.1.1	Neighbor rules for IPV4/6.

2.1854.1.1.1 NeighborRule

[destinationPort]	*integer*	Destination port.
[sourcePort]	*integer*	Source port.
sourcePrefix	*IP address prefix*	Source prefix.

2.1855 show transceiver application [...]

ports	*table(string) = TransceiverApplications* see 2.1855.1	Mapping between port name and transceiver applications.

2.1855.1 TransceiverApplications

applications	*table(numeric string) = TransceiverApplication-Status* see 2.1855.1.1	Mapping between application IDs and the information associated with each application.

2.1855.1.1 TransceiverApplicationStatus

applicationCode	*integer*	Indicates the active application code.
hostInterface	*string*	Indicates the host interface of the active application.

hostLanes	*table(numeric string) = TransceiverApplicationHostLanes* see 2.1855.1.1.1	Mapping between application start lane and all the lanes under the given application.
mediaInterface	*string*	Indicates the media interface of the active application.
statusCodes	*table(string) = TransceiverApplicationHostLanes* see 2.1855.1.1.1	Mapping between configuration error code and list of lanes that report the given code.

2.1855.1.1.1 TransceiverApplicationHostLanes

lanes	*list of integer*	Host lanes (indexed starting at 1).

2.1856 show transceiver error [...]

ports	*table(string) = TransceiverErrorValues* see 2.1856.1	Mapping between port name and transceiver errors.

2.1856.1 TransceiverErrorValues

transceiverErrors	*table(string) = TransceiverErrorValueFormat* see 2.1856.1.1	A mapping of error to its details.

2.1856.1.1 TransceiverErrorValueFormat

count	*integer*	Number of failures since system boot.
lastError	*floating point number*	Last error update time in UTC.

[lastReason]	*enum: moduleFaultCauseUnknown | smbusErrEepromRead | moduleFaultCauseData-MemCorrupted | moduleFaultCauseTec-Runaway | smbusErrNone | smbusErrUnspecified | smbusErrMonitor | configAccepted | configNoStatus | configRejected | smbusErrFaulty | faultCauseSmbusError | moduleNoFaultCause | faultCauseUnsupported-Xcvr | faultCauseInvalidChecksum | smbusErrConfig | moduleFaultCauseCustom | faultCauseNone | moduleFaultCause-ProgramMemCorrupted | faultCauseUnsupported-Power | moduleFaultCauseReserved | smbusErrTestPattern | smbusErrInitialization*	Last reason for error.

2.1857 show transceiver error-correction coherent degrade [...]

slots	*table(Interface name) = TransceiverCoherentFec-DegradeSlot* see 2.1857.1	Mapping between slot name and coherent error-correction degrade info.

2.1857.1 TransceiverCoherentFecDegradeSlot

[detected]	*TransceiverCoherentFec-DegradeParameter* see 2.1857.1.1	Detected degrade status.
[excessive]	*TransceiverCoherentFec-DegradeParameter* see 2.1857.1.1	Excessive degrade status.

2.1857.1.1 TransceiverCoherentFecDegradeParameter

[configured]	*TransceiverCoherentFec-DegradeThresholds* see 2.1857.1.1.1	Values of configured degrade thresholds.
[operational]	*TransceiverCoherentFec-DegradeThresholds* see 2.1857.1.1.1	Values of operational degrade thresholds.

status	*enum: notPresent | ok | unknown | alarm | unsupported*	Error-correction degrade alarm status.

2.1857.1.1.1 TransceiverCoherentFecDegradeThresholds

tClear	*floating point number*	Value of error-correction degrade clear threshold.
tRaise	*floating point number*	Value of error-correction degrade raise threshold.

2.1858 show transceiver squelch [...]

slots	*table(string) = TransceiverSquelchTypes* see 2.1858.1	Mapping between slot name and transceiver squelch types.

2.1858.1 TransceiverSquelchTypes

rxAutoSquelch	*TransceiverSquelchType* see 2.1858.1.1	Transceiver RX auto squelch.

2.1858.1.1 TransceiverSquelchType

[configured]	*enum: disabled | moduleDefault | enabled*	The user-configured squelch setting.
lanes	*table(numeric string) = TransceiverLaneSquelch-Setting* see 2.1858.1.1.1	Mapping from lane to transceiver squelch setting.
supported	*boolean*	Configuration supported on this module.

2.1858.1.1.1 TransceiverLaneSquelchSetting

[default]	*boolean*	Module default setting.
operational	*boolean*	Module operational setting.

2.1859 show transceiver status [...]

see 2.858

2.1860 show transceiver status firmware update [...]

ports	*table(string) = TransceiverFirmwareUpdateSlotDetail* see 2.1860.1	Mapping between the slot name and the firmware update information of the inserted transceiver.

2.1860.1 TransceiverFirmwareUpdateSlotDetail

[firmwareUpdate]	*TransceiverFirmwareUpdateStatus* see 2.1860.1.2	Firmware update status.
[hardwareDependentCommand]	*TransceiverFirmwareHardwareStatus* see 2.1860.1.1	Hardware-level firmware update command status.
serialNumber	*string*	Transceiver serial number.
serialNumberLastChange	*floating point number*	UTC timestamp of the last change.

2.1860.1.1 TransceiverFirmwareHardwareStatus

cdbCommandAckTimeout	*TransceiverFirmwareCdbTimeout* see 2.1860.1.1.1	Number of CDB command timeouts.
commandType	*enum: start | download | run | commit | abort | complete*	Last CDB command issued.
commandTypeLastChange	*floating point number*	UTC timestamp of the last change.
status	*enum: in progress | success | unknown | fail*	Last CDB command status.
[statusDetail]	*TransceiverFirmwareHardwareStatusDetail* see 2.1860.1.1.2	Details for the last CDB result.
statusLastChange	*floating point number*	UTC timestamp of the last change.

2.1860.1.1.1 TransceiverFirmwareCdbTimeout

lastChange	*floating point number*	UTC timestamp of the last change.
timeoutCount	*integer*	Number of CDB command timeouts.

2.1860.1.1.2 TransceiverFirmwareHardwareStatusDetail

[decoded]	*string*	Result of the last CDB command.
lastChange	*floating point number*	UTC timestamp of the last change.
[rawValue]	*integer*	Raw status code of the last CDB commands.

2.1860.1.2 TransceiverFirmwareUpdateStatus

downloadPercentage	*integer*	Firmware update progress.
downloadPercentageLastChange	*floating point number*	UTC timestamp of the last change.

restartPercentage	*integer*	Firmware restart progress.
restartPercentageLastChange	*floating point number*	UTC timestamp of the last change.
status	*enum: idle | running | terminated | success | failed | timeout*	Firmware update status.
statusLastChange	*floating point number*	UTC timestamp of the last change.

2.1861 show transceiver status firmware version [...]

[ports]	*table(string) = TransceiverFirmwareCollection* see 2.1861.1	Mapping between the slot name and the firmware version information of the inserted transceiver.

2.1861.1 TransceiverFirmwareCollection

firmware	*TransceiverFirmwareVersionDetailDict* see 2.1861.1.1	images present on the transceiver.

2.1861.1.1 TransceiverFirmwareVersionDetailDict

active	*table(string) = TransceiverFirmwareVersionDetail* see 2.1861.1.1.1	Mapping from an active image to the image details.
inactive	*table(string) = TransceiverFirmwareVersionDetail* see 2.1861.1.1.1	Mapping from an inactive image to the image details.

2.1861.1.1.1 TransceiverFirmwareVersionDetail

buildNum	*integer*	Firmware build number.
committed	*boolean*	This firmware image will become active upon transceiver reinsertion.
majorVer	*integer*	Firmware major version.
minorVer	*integer*	Firmware minor version.

2.1862 show transceiver traffic-loopback capabilities [...]

loopbackCaps	*table(Interface name) = Capabilities* see 2.1862.1	Loopback capabilities of each transceiver that supports CMIS based loopback.

2.1862.1 Capabilities

capabilities	*list of Capability* see 2.1862.1.1	Loopback capability set.
simultaneousHostMedia	*boolean*	Module supports simultaneous loopback configuration on both the host and the media side.

2.1862.1.1 Capability

kind	*enum: system | perLane | network*	The type of loopback capability, either indicates traffic source or the per-lane capability.
side	*enum: host | media*	The side of the transceiver on which the loopback is considered on.
supported	*boolean*	Loopback capability is supported.

2.1863 show transceiver traffic-loopback status [...]

transceiverLoopback	*table(Interface name) = SlotInfo* see 2.1863.1	Loopback configuration and status of each transceiver that supports CMIS based loopback.

2.1863.1 SlotInfo

hostLanes	*table(numeric string) = LaneInfo* see 2.1863.1.1	Host side lane configuration and status information, keyed by lane ID.
mediaLanes	*table(numeric string) = LaneInfo* see 2.1863.1.1	Media side lane configuration and status information, keyed by lane ID.

2.1863.1.1 LaneInfo

configured	*enum: system | none | network*	The loopback source configuration.
operational	*enum: notPresent | network | system | notSupported | none*	The operational state of the loopback source configuration.

2.1864 show tunnel counters ip-in-ip [...]

entries	*list of TunnelCountersEntry* see 2.1864.1	tunnel hardware counters.

2.1864.1 TunnelCountersEntry

[endpoint]	*IP address prefix*	Endpoint of the tunnel.
tunnelIndex	*integer*	Index of the tunnel entry in the source tunnel table.
[tunnelName]	*string*	Name for the tunnel entry.
[txBytes]	*integer*	Total number of transmitted bytes.
[txPackets]	*integer*	Total number of transmitted packets.

2.1865 show tunnel counters mpls rate [...]

entries	*list of MplsTunnelCounters-RateEntry* see 2.1865.1	MPLS tunnel hardware rates.

2.1865.1 MplsTunnelCountersRateEntry

[endpoint]	*IP address prefix*	Endpoint of the tunnel.
tunnelIndex	*integer*	Index of the tunnel entry in the source tunnel table.
[tunnelName]	*string*	Name for the tunnel entry.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.
[txBps]	*floating point number*	Bits per second.
[txPps]	*floating point number*	Packets per second.

2.1866 show tunnel fib [...]

revision 2

categories	*table(string) = TunnelFibCategory* see 2.1866.1	Collections of tunnels in the FIB, grouped by tunnel service type.

2.1866.1 TunnelFibCategory

entries	*table(numeric string) = TunnelFibEntry* see 2.1866.1.1	Tunnel FIB entries for a tunnel type.

2.1866.1.1 TunnelFibEntry

[algorithm] **	*string*	Algorithm associated with the endpoint. **Since: 4.26.0
[aliasEndpoints] **	*list of IP address prefix*	Additional prefixes used for underlay nexthop resolution. **Since: 4.26.0
backupVias	*list of TunnelFibVia* see 2.151.1.4	List of tunnel backup vias.
[endpoint]	*IP address prefix*	Route prefix used for underlay nexthop resolution.
[interface]	*string*	Interface name for tunnel.
[seqNo] **	*integer*	Sequence number of tunnel entry. **Since: 4.24.1
[tunnelFecId]	*integer*	FEC identifier.
[tunnelId]	*integer*	Tunnel identifier.
tunnelIndex	*integer*	Tunnel index within tunnel type.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Nexthop | IS-IS SR | EosSdk sBFD | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | RSVP FRR | RSVP LER | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.
[tunnelViaStatus]	*enum: unresolved | usingBackupVias | unknown | usingPrimaryVias | notProgrammed*	Tunnel programming status.
vias	*list of TunnelFibVia* see 2.151.1.4	List of tunnel vias.

2.1867 show tunnel fib multicast [...]

revision 2

see 2.1866

2.1868 show tunnel rib (system-tunneling-ldp-tunnel-rib | system-igp-shortcut-tunnel-rib | system-tunnel-rib | TUNNEL_RIB_NAME) summary

name	*string*	Tunnel RIB name.
tunnels	*table(string) = integer*	Tunnel count keyed by tunnel type.

2.1869 show tunnel rib brief [...]

revision 3

entries	*table(IPv4 or IPv6 address prefix) = TunnelRibEntries-ByTunnelType* see 2.1869.1	Tunnel entries keyed by prefix.
name	*string*	Tunnel RIB name.

2.1869.1 TunnelRibEntriesByTunnelType

entries	*list of TunnelRibEntry* see 2.1869.1.1	List of tunnel entries of different tunnel types towards the same endpoint.

2.1869.1.1 TunnelRibEntry

metric	*integer*	IGP metric of tunnel endpoint entry.
metric2	*integer*	IGP metric 2 of tunnel endpoint entry.
[metricType] **	*enum: MED | metric | AIGP*	Metric type of tunnel endpoint entry. **Since: 4.26.0
pref	*integer*	Underlying IGP route's preference.
pref2	*integer*	Preference 2 of tunnel endpoint entry.
tunnelIndexes	*list of integer*	List of tunnel indexes ending at this tunnel endpoint.
tunnelPreference	*integer*	Tunnel preference used for tunnel RIB election.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.1870 show tunnel rib candidates [...]

endpoints	*table(IPv4 or IPv6 address prefix) = InternalTunnelRibCandidates* see 2.1870.1	Candidate tunnel entries, keyed by prefix.
name	*string*	Tunnel RIB name.

2.1870.1 InternalTunnelRibCandidates

candidates	*list of InternalTunnelRibCandidateEntry* see 2.1870.1.1	Candidate entries for tunnel RIB election, sorted by preference order.

2.1870.1.1 InternalTunnelRibCandidateEntry

tunnelIndexes	*list of integer*	Indexes of tunnels ending at this tunnel endpoint.
tunnelMetric	*integer*	Tunnel metric used for tunnel RIB election.
tunnelPreference	*integer*	Tunnel preference used for tunnel RIB election.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.1871 show tunnel rib colored system-colored-tunnel-rib brief [...]

endpoints	*table(IPv4 or IPv6 address prefix) = TunnelRibColors* see 2.1871.1	Colored tunnel entries keyed by endpoint.
name	*string*	System colored tunnel RIB name.

2.1871.1 TunnelRibColors

colors	*table(numeric string) = TunnelRibEntry* see 2.1869.1.1	Colored tunnel entries keyed by color.

2.1872 show tunnel rib multicast (mldp | static)

name	*string*	Tunnel RIB name.
tunnels	*table(string) = MulticastTunnelRibEntry* see 2.1872.1	Tunnel entries keyed by FEC Element.

2.1872.1 MulticastTunnelRibEntry

tunnelIndexes	*list of integer*	List of tunnel indexes ending at this tunnel endpoint.
tunnelType	*enum: RSVP | VXLAN VTEP Encap | Static | Nexthop | IS-IS SR | EosSdk sBFD | Static IPv6 | Static MPLS Multicast | IS-IS FlexAlgo | Nexthop Group | Static MPLS | VXLAN | maxTunnelType | IS-IS SR IPv6 | RSVP FRR | RSVP LER | IS-IS SR IPv4 | SR-TE | invalidTunnel | SR-TE Policy | BGP LU | EVPN VXLAN | MLDP P2MP | LDP | Static Interface | IP-in-IP | TI-LFA | SRv6 Transport | BGP LU Forwarding | RSVP LER SUB*	Tunnel type.

2.1873 show tunnel rib summary

tunnels	*table(string) = integer*	Tunnel count keyed by tunnel RIB name.

2.1874 show upnp counters [...]

vrfs	*table(string) = VrfCounters* see 2.1874.1	UPNP counters keyed by VRF name.

2.1874.1 VrfCounters

addConflicts	*integer*	UPNP flow conflicts.
addPrematureSessionClosure	*integer*	Premature session closures.
addRefused	*integer*	NAT flow conflicts.
addRequestsTotal	*integer*	Total AddPortMapping requests.
addSuccess	*integer*	Successful AddPortMapping requests.
delFailed	*integer*	DeletePortMapping failures.
delRequestsTotal	*integer*	Total DeletePortMapping requests.
delSuccess	*integer*	Successful DeletePortMapping requests.
getSpecificMappingFailed	*integer*	GetSpecificMapping failures.

getSpecificMappingRequests-Success	*integer*	Successful GetSpecificMapping requests.
getSpecificMappingRequests-Total	*integer*	Total GetSpecificMapping requests.
portMappingClearedLocally	*integer*	Port mappings cleared locally.

2.1875 show upnp igd port-map

vrfs	*table(string) = VrfUpnpPortMap* see 2.1875.1	Map of VRF Names to the UPnP port map.

2.1875.1 VrfUpnpPortMap

portMaps	*list of UpnpPortMapEntry* see 2.1875.1.1	UPnP port maps.

2.1875.1.1 UpnpPortMapEntry

externalIpAndPort	*IpAddrAndPort* see 2.616.1.1.1.1	Global IP address and port.
internalIpAndPort	*IpAddrAndPort* see 2.616.1.1.1.1	IP address of the client and port.
leaseTime	*floating point number*	Lease time in seconds.
protocol	*integer*	IP protocol.
[state]	*enum: refused | adding | deleting | NA | created | cleared*	Port map state.

2.1876 show upnp service

vrfs	*table(string) = UpnpServiceEntry* see 2.1876.1	Map of VRF names to UPnP service status.

2.1876.1 UpnpServiceEntry

ssdpStatus	*enum: admin down | down | up*	UPnP service status.
upnpStatus	*enum: admin down | down | up*	UPnP service status.

2.1877 show uptime

see 2.1587.2

2.1878 show user-account

`users`	*table(string) = User* see 2.1878.1	Maps a username to details for this user.

2.1878.1 User

`privLevel`	*integer*	The configured privilege level for this user (between 0 and 15 inclusive).
`[role]`	*string*	What role the account has been given (e.g. 'network-admin').
`[secondarySshKeys]` **	*list of string*	SSH secondary keys for this user. **Since: 4.28.2
`[sshAuthorizedKey]`	*string*	SSH public key for this user.
`username`	*string*	The username of this account.

2.1879 show users

`serials`	*table(string) = ShowAaaUserInfo* see 2.1879.1	Terminal information for serial connection.
`vtys`	*table(string) = ShowAaaUserInfo* see 2.1879.1	Terminal information for ssh or telnet connections.

2.1879.1 ShowAaaUserInfo

`idleTime`	*integer*	Duration of session's inactivity in seconds.
`lineNumber` **	*integer*	Line number of the user session. **Since: 4.25.2
`[location]`	*string*	Source IP address or hostname for vty sessions and absent for serial connections.
`sessionTty`	*boolean*	Indicates if this is the terminal used by the session that just ran this command.
`user`	*string*	Login name of the user of this session.

2.1880 show users accounts

see 2.1878

2.1881 show users detail

see 2.67

2.1882 show users roles [...]

see 2.1673

2.1883 show version [...]

architecture	*string*	Control plane CPU architecture.
bootupTimestamp	*floating point number*	Bootup timestamp of the system.
[cEosToolsVersion] **	*string*	cEOS tools version number. **Since: 4.20.0
[configMacAddress] **	*MAC address*	Configured MAC address of the system, change takes effect only after reboot. Available only on fixed systems. **Since: 4.24.0
[details]	*Details* see 2.1883.1	Detailed version information.
hardwareRevision	*string*	Name of the hardware revision of the switch.
[hwMacAddress] **	*MAC address*	Hardware MAC address of the system (might not be in use if there is a configured MAC). **Since: 4.24.0
imageFormatVersion **	*string*	EOS image format version number. **Since: 4.26.2
imageOptimization **	*string*	Optimization name of squash filesystem booted by the switch. **Since: 4.27.2
internalBuildId	*string*	Unique internal build ID.
internalVersion	*string*	Full internal EOS version number.
isIntlVersion **	*boolean*	Whether the running image is an international EOS image. **Since: 4.17.0
[kernelVersion] **	*string*	Kernel version. **Since: 4.25.0
memFree	*integer*	Available free memory in KB.
memTotal	*integer*	Size of the memory in KB.
mfgName **	*string*	Manufacturer name of the switch. **Since: 4.21.3
modelName	*string*	Model name of the switch.
serialNumber	*string*	Serial number of the switch.
systemMacAddress	*MAC address*	The operational MAC address of the system.
uptime **	*floating point number*	Uptime of the system. **Since: 4.21.0
version	*string*	EOS software image version.

2.1883.1 Details

components	*list of Component* see 2.1883.1.2	Hardware component versions.
deviations	*list of string*	List of hardware deviations.
[onieVersion] **	*string*	Version of onie-installer used to provision EOS on the switch. **Since: 4.22.0
packages	*table(string) = Package* see 2.1883.1.1	RPM packages installed on the switch, keyed by package name.

switchType	*enum: chassis* | *fixedSystem* | *unknown*	The physical type of switch. Is 'unknown' if the system has not initialized.
systemEpoch **	*string*	Version of the system epoch. **Since: 4.17.0

2.1883.1.1 Package

release	*string*	Release string of the RPM package.
version	*string*	Version string of the package.

2.1883.1.2 Component

name	*string*	Name of the hardware component.
version	*string*	Version of the hardware component.

2.1884 show version compatibility SWI

errorList	*list of string*	List of compatibility blocking errors.
image	*string*	Software image URL.
warningList	*list of string*	List of compatibility warnings.

2.1885 show version firmware

Could also be an instance of:
Details 2.1883.1

components	*list of Component* see 2.1883.1.2	Hardware component versions.
switchType	*enum: chassis* | *fixedSystem* | *unknown*	The physical type of switch. Is 'unknown' if the system has not initialized.

see 2.1883.1

2.1886 show version license

licenses	*table(string)* = *string*	Licenses covering software present in EOS.

2.1887 show vlan [...]

[sourceDetail]	*string*	A description of the source of this configuration. When using MLAG, this field may indicate that the primary peer provided a config that overrides the secondary's local config.
vlans	*table(numeric string)* = *Status* see 2.1887.1	Configured VLANs, keyed by VLAN Id.

2.1887.1 Status

dynamic	*boolean*	Indicates whether or not this VLAN is dynamic.
interfaces	*table(string) = Interface* see 2.1887.1.1	List of interfaces that are a member of this VLAN, keyed by interface name.
name	*string*	VLAN name.
status	*enum: suspended | inactive | active*	VLAN administrative status.

2.1887.1.1 Interface

[annotation]	*string*	Annotation explaining any configuration specific to this interface, i.e. that the interface is in a port-channel.
blocked **	*boolean*	Traffic is being blocked on this interface. **Since: 4.26.0
[function] **	*string*	Use of this interface. **Since: 4.26.2
[functionDetail] **	*string*	Use of this interface. **Since: 4.26.2
privatePromoted	*boolean*	This is a private VLAN promoted port.

2.1888 show vlan brief [...]

see 2.1887

2.1889 show vlan brief count

[sourceDetail]	*string*	A description of the source of this configuration. When using MLAG, this field may indicate that the primary peer provided a config that overrides the secondary's local config.
totalVlans	*integer*	Number of existing VLANs.

2.1890 show vlan counters [...]

vlanCountersInfo	*table(numeric string) = VlanStats* see 2.1890.1	A mapping between VLAN ID and counter data.

2.1890.1 VlanStats

inBcastPkts	*integer*	Input broadcast packets.
inMcastPkts	*integer*	Input multicast packets.
inOctets	*integer*	Input octets.
inUcastPkts	*integer*	Input unicast packets.
outBcastPkts	*integer*	Output broadcast packets.

outMcastPkts	*integer*	Output multicast packets.
outOctets	*integer*	Output octets.
outUcastPkts	*integer*	Output unicast packets.

2.1891 show vlan dynamic

dynamicVlans	*table(string) = VlanIds* see 2.1891.1	A Mapping between the Dynamic VLAN source and the VLANS.

2.1891.1 VlanIds

Could also be an instance of:
TrunkVlans 2.510.1.1

vlanIds	*list of integer*	A list of the VLAN Ids.

2.1892 show vlan interface dynamic

agents	*table(string) = VlanIntfs* see 2.1892.1	A Mapping between the Dynamic VLAN interface source and the VLANS.

2.1892.1 VlanIntfs

interfaces	*list of Interface name*	A list of the VLAN interfaces.

2.1893 show vlan internal allocation policy

endVlanId	*integer*	Ending VLAN Id.
policy	*enum: ascending | descending*	Internal VLAN allocation policy.
startVlanId	*integer*	Starting VLAN Id.

2.1894 show vlan internal usage

internalVlans	*table(numeric string) = string*	Mapping between the VLAN Id and the VLAN Name.

2.1895 show vlan internal usage reserved

internalVlans	*table(numeric string) = string*	Mapping between the VLAN Id and the corresponding Interface Name.

2.1896 show vlan private-vlan

privateVlans	*table(numeric string) = PrivateVlans* see 2.1896.1	A mapping between a primary VLAN Id and a list of private VLAN information.

2.1896.1 PrivateVlans

privateVlan	*list of PrivateVlan* see 2.1896.1.1	A list of the private VLAN information.

2.1896.1.1 PrivateVlan

interfaces	*list of Interface name*	List of interfaces in VLAN.
secondaryVlanId	*integer*	The secondary VLAN Id.
vlanType	*enum: isolated | normal | community*	The type of Private VLAN.

2.1897 show vlan summary

see 2.1889

2.1898 show vlan trunk group [...]

[sourceDetail]	*string*	A description of the source of this configuration. When using MLAG, this field may indicate that the primary peer provided a config that overrides the secondary's local config.
trunkGroups	*table(numeric string) = GroupNames* see 2.1898.1	Trunk groups associated with each VLAN.

2.1898.1 GroupNames

names	*list of string*	Trunk group names.

2.1899 show vmtracer all [...]

revision 2

sessions	*table(string) = VmTracerSwitches* see 2.1899.1	A mapping of session names to vCenter switches.

2.1899.1 VmTracerSwitches

switches	*table(string) = SwitchInterfaces* see 2.1899.1.1	A mapping of switch names to switch interface sets.

2.1899.1.1 SwitchInterfaces

intfs	*table(string) = VmTracerInterface* see 2.1899.1.1.1	A mapping of interface names to interfaces.

2.1899.1.1.1 VmTracerInterface

hostName	*string*	Host name where this interface is attached.
switchName	*string*	Vswitch this interface is attached to.
uplinkName	*string*	Uplink port this interface is attached to.
vms	*table(MAC address) = VmTracerVm* see 2.1899.1.1.1.1	A mapping between VM mac and VM information.

2.1899.1.1.1.1 VmTracerVm

datacenter	*string*	Name of the Data Center where the VM resides.
[failoverToleranceState]	*enum: ftDisabled | ftStandby | ftActive*	Failover tolerance state.
host	*string*	Host name where VM resides.
[inVmotion]	*boolean*	Flag indicating if a VM is being Vmotioned.
interface	*string*	Interface on which the VM is connected.
[logicalSwitch]	*string*	Name of Vmware logical switch this vm is connected to.
macAddress	*MAC address*	MAC address of the VM.
[macStatus]	*enum: unlearnt | learnt*	VM Mac address learn status.
[nsxConnected]	*boolean*	Flag indicating if an NSX connection is active.
[physicalNic]	*string*	Name of the physical nic the VM is attached to.
portGroup	*string*	Name of the port group to which the VM belongs.
powerStatus	*enum: down | up*	Power status of the VM.
switch	*string*	Name of the vSwitch or DVS the VM is on.
[switchMtu]	*integer*	MTU of the vSwitch or DVS this VM is on.

[transportVlan]	*integer*	Transport VLAN for VXLAN VTEP.
[transportZone]	*string*	VXLAN Transport Zone of this VM.
[vCenterUrl]	*string*	Url of associated vCenter instance.
vNic	*string*	Name of the Vm Nic.
vlan	*string*	Portgroup vlan as 'native', 'trunk' or string representation of a vlan.
vmName	*string*	Name of the VM.
[vmType]	*enum: vmStandard | vmKernel*	Type of VM entry.
[vni]	*integer*	VNI of VXLAN that this VM is attached to.
[vtepIp]	*IPv4 address*	IP address of the VTEP used by this vm.

2.1900 show vmtracer interface (ETH | ETHS | LAG) vm VMNAME

intfs	*table(Interface name) = VmTracerGenericInterface* see 2.1900.1	A mapping of interface names to interfaces.

2.1900.1 VmTracerGenericInterface

Could also be an instance of:
VmTracerInterface 2.1899.1.1.1
VmTracerLagInterface 2.1900.2

vms	*table(MAC address) = VmTracerVm* see 2.1899.1.1.1.1	A mapping between VM mac and VM information.

2.1900.2 VmTracerLagInterface

members	*table(Interface name) = VmTracerInterface* see 2.1899.1.1.1	A mapping between interface names and LAG members.
vms	*table(MAC address) = VmTracerVm* see 2.1899.1.1.1.1	A mapping between VM mac and VM information.

2.1901 show vmtracer interface [...]

see 2.1900

2.1902 show vmtracer logical-switch [...]

scopes	*table(string) = NetworkScope* see 2.1902.1	A mapping of network scope display names to scopes.

2.1902.1 NetworkScope

[desc] **	*string*	Description of the network scope. **Since: 4.20.0
switches	*table(string) = LogicalSwitch* see 2.1902.1.1	A mapping of logical switch ids to switches.

2.1902.1.1 LogicalSwitch

controlPlane	*ControlPlane* see 2.1902.1.1.1	Mode of VMWare control plane.
displayName	*string*	Logical switch display name.
vni	*integer*	VXLAN VNI.

2.1902.1.1.1 ControlPlane

[mcastAddr]	*IPv4 address*	VXLAN ARP broadcast address.
mode	*enum: hybrid | multicast | unicast*	NSX mode.

2.1903 show vmtracer session SESSION (vcenter | vsm) [...]

sessionStatuses	*table(string) = VmTracerSessionStatus* see 2.1903.1	A mapping of session names to status information.

2.1903.1 VmTracerSessionStatus

[vCenterStatus]	*VCenterServiceStatus* see 2.1903.1.1	vCenter status.
[vShieldStatus]	*VShieldServiceStatus* see 2.1903.1.2	vShield manager status.

2.1903.1.1 VCenterServiceStatus

allowedVlans	*VlanIds* see 2.1891.1	List of VLANs allowed in the vCenter session.
autoVlan	*boolean*	Whether auto VLAN provision is enabled.
serviceStatus	*WebServiceStatus* see 2.1903.1.1.1	vCenter appliance service status.

2.1903.1.1.1 WebServiceStatus

[details]	*StatusDetail* see 2.1903.1.1.1.1	Details of connection status associated with vShield/vCenterthat is only populated when requested by cli command.
sessionState	*enum: stateUnknown | stateConnecting | stateDisconnected | stateConnected*	Current vCenter/vShield session state.
sourceIntf **	*interface name*	Local interface for vCenter/vShield connections. **Since: 4.21.3
url	*string*	vCenter/vShield/NSX-V URL.
username	*string*	vCenter/vShield username.
vrf **	*string*	vCenter/vShield VRF. **Since: 4.21.3
vrfStatus **	*enum: vrfStatusReady | vrfStatusUnknown | vrfStatusError | vrfStatusWaiting*	VRF state of current session. **Since: 4.21.3

2.1903.1.1.1.1 StatusDetail

lastErrorCode	*string*	The last error code.
lastFailedMsg	*string*	The last failed message.
lastFailedMsgTime	*floating point number*	The time of the last failed message.
lastMsgSent	*string*	The last message sent.
lastStateChangeTime	*floating point number*	The time of last state change.
lastSuccessfulMsg	*string*	The last successful message.
lastSuccessfulMsgTime	*floating point number*	The time of the last successful message.
numFailedMsg	*integer*	The number of failed messages.

numSuccessfulMsg	*integer*	The number of successful messages.
responseTimeForLastMsg	*floating point number*	The time it took, in seconds, to send response for the last message.
timeOfLastMsg	*floating point number*	The time of the last message.

2.1903.1.2 VShieldServiceStatus

serviceStatus	*WebServiceStatus* see 2.1903.1.1.1	vShield appliance service status.

2.1904 show vmtracer session [...]

see 2.1903

2.1905 show vmtracer vm VM_NAME

revision 2

staleSessions	*table(string) = floating point number*	A mapping of stale session names to their sync time.
vms	*list of VmTracerVm* see 2.1899.1.1.1.1	List of VMs.

2.1906 show vmtracer vm [...]

revision 2

see 2.1905

2.1907 show vmtracer vxlan segment [...]

see 2.1902

2.1908 show vmtracer vxlan segment range

mcastRanges	*list of McastRange* see 2.1908.1	Multicast range.
vniRanges	*list of VniRange* see 2.1908.2	Vni range.

2.1908.1 McastRange

begin	*IPv4 address*	Start of Multicast Range.
end	*IPv4 address*	End of Multicast Range.

2.1908.2 VniRange

begin	*integer*	Start of VNI range.
end	*integer*	End of VNI range.

2.1909 show vpls [...]

instances	*table(string) = VplsInstance* see 2.1909.1	VPLS instance information keyed by instance name.

2.1909.1 VplsInstance

macWithdrawalTrigger-Interface	*boolean*	Send MAC withdrawal messages on instance pseudowires when VPLS local interface goes down.
pseudowireGroups	*table(string) = VplsPseudowireGroup* see 2.1909.1.1	VPLS pseudowire group information keyed by group name.
vlans	*table(numeric string) = VplsVlan* see 2.1909.1.2	VPLS VLAN information keyed by VLAN ID.

2.1909.1.1 VplsPseudowireGroup

[forwardingPreferenceEvpn] **	*boolean*	Prefer EVPN control plane for data forwarding as per RFC8560. **Since: **currentRelease**
macWithdrawalPropagation	*enum: locallyTriggered | suppress | locallyTriggered-AndReceived*	Propagate locally triggered and received MAC withdrawal messages.
macWithdrawalTriggerOn-PseudowireFailure	*boolean*	Trigger MAC withdrawal transmission when a group pseudowire state transitions away from up.
neighbors	*table(IPv4 or IPv6 address) = VplsPseudowireGroup-Neighbor* see 2.1909.1.1.1	VPLS neighbor pseudowire information keyed by neighbor address.
[profile] **	*string*	VPLS auto-discovery LDP pseudowire profile configuration. **Since: 4.28.0
splitHorizon	*boolean*	VPLS split-horizon rule enforced across group pseudowires.

2.1909.1.1.1 VplsPseudowireGroupNeighbor

pseudowires	*table(string) = VplsPseudowire* see 2.1909.1.1.1.1	VPLS pseudowire information keyed by pseudowire ID.

2.1909.1.1.1.1 VplsPseudowire

[controlWord]	*boolean*	Control word used.
[flowLabelUsed]	*enum: disabled | both | transmit | receive*	Flow label used.
[interface]	*interface name*	Pseudowire interface.
[lastNegativeMacWithdrawal-ReceivedTime]	*floating point number*	Time when last negative MAC withdrawal all-from-me message was received.

[lastNegativeMacWithdrawal-SentTime]	*floating point number*	Time when last negative MAC withdrawal all-from-me message was sent.
[localConfig]	*LdpPseudowirePeerConfig* see 2.1175.1.1.2.3	Local pseudowire configuration.
[localStatus]	*enum: up* | *standby*	Local pseudowire status.
[negativeMacWithdrawal-Received]	*integer*	Number of negative MAC withdrawal all-from-me messages received.
[negativeMacWithdrawalSent]	*integer*	Number of negative MAC withdrawal all-from-me messages sent.
[neighborConfig]	*LdpPseudowirePeerConfig* see 2.1175.1.1.2.3	Neighbor pseudowire configuration.
[neighborStatus]	*enum: noRemoteEnd* | *controlWordMismatch* | *noLocalMapping* | *pwTypeMismatch* | *standby* | *pwNotForwardingRemote* | *down* | *mtuMismatch* | *up* | *noPeerLabel*	Neighbor pseudowire status.
[pseudowireType]	*enum: pwType5* | *pwType4*	Pseudowire type.
status	*enum: noRemoteEnd* | *unsupportedTunnelMode* | *noTunnelRib* | *controlWordMismatch* | *neighborEvpn* | *noLocalMapping* | *pwTypeMismatch* | *standby* | *noTunnel* | *pwNotForwardingRemote* | *down* | *cliConflict* | *cliIncomplete* | *mtuMismatch* | *up* | *noPeerLabel*	Pseudowire status.
[tunnelIndexes]	*list of integer*	List of tunnel indexes ending at this tunnel endpoint.
[tunnelType]	*enum: RSVP* | *VXLAN VTEP Encap* | *Static* | *Nexthop* | *IS-IS SR* | *EosSdk sBFD* | *Static IPv6* | *Static MPLS Multicast* | *IS-IS FlexAlgo* | *Nexthop Group* | *Static MPLS* | *VXLAN* | *maxTunnelType* | *IS-IS SR IPv6* | *RSVP FRR* | *RSVP LER* | *IS-IS SR IPv4* | *SR-TE* | *invalidTunnel* | *SR-TE Policy* | *BGP LU* | *EVPN VXLAN* | *MLDP P2MP* | *LDP* | *Static Interface* | *IP-in-IP* | *TI-LFA* | *SRv6 Transport* | *BGP LU Forwarding* | *RSVP LER SUB*	Tunnel type.

2.1909.1.2 VplsVlan

[normalizedTag]	*integer*	Pseudowire VPLS 802.1Q tag value.

2.1910 show vpls pseudowire brief (all | { (neighbor NEIGHBOR) | (pseudowire-type PW_TYPE) }) [...]

see 2.1909

2.1911 show vrf [...]

vrfs	*table(string) = VrfEntry* see 2.1911.1	Mapping of Vrf name to Vrf information.

2.1911.1 VrfEntry

interfaces	*list of Interface name*	List of interfaces which are assigned to this vrf.
[interfacesV4] **	*list of Interface name*	List of IPv4 interfaces which are assigned to this vrf. **Since: 4.26.2
[interfacesV6] **	*list of Interface name*	List of IPv6 interfaces which are assigned to this vrf. **Since: 4.26.2
protocols	*IpProtocolDesc* see 2.1911.1.1	Information on ipv4 and ipv6 support for this vrf.
routeDistinguisher	*string*	Route Distinguisher associated with this vrf.
vrfState	*enum: creating | deleting | up*	Various states of vrf. 'creating' indicates vrf is being created, 'up' indicates vrf is created successfully and 'deleting' indicates vrf is being deleted.

2.1911.1.1 IpProtocolDesc

ipv4	*ProtocolInfo* see 2.1911.1.1.1	ipv4 protocol information for a vrf.
ipv6	*ProtocolInfo* see 2.1911.1.1.1	ipv6 protocol information for a vrf.

2.1911.1.1.1 ProtocolInfo

[multicastState]	*enum: initializing | up*	Indicates the state of the multicast routing after it is configured.
protocolState	*enum: down | up*	Protocol state is down when the vrf is being created, state is up otherwise.

routingState	*enum: initializingUp | down | up | initializingDown*	Indicates the various states of routing for the Vrf.'up' indicates routing is enabled for this vrf. 'down' indicates routing is disabled for this vrf. 'initializing up/down' indicates routing state is being configured.
supported	*boolean*	Indicates whether the protocol is supported.

2.1912 show vrf leak connected (ipv4 | ipv6)

destVrfs	*table(string) = LeakedConnectedRoutes* see 2.1912.1	A mapping of destination VRF to leaked connected routes.

2.1912.1 LeakedConnectedRoutes

routes	*table(IPv4 or IPv6 address prefix) = string*	A mapping of route prefix to the VRF that leaked it.

2.1913 show vrf leak id

vrfs	*table(string) = integer*	A mapping between VRF name and VRF ID.

2.1914 show vrf reserved-ports [...]

vrfs	*table(string) = VrfReservedPortsEntry* see 2.1914.1	Mapping of VRF name to its reserved ports.

2.1914.1 VrfReservedPortsEntry

config	*string*	Reserved Ports configuration associated with this VRF.
status	*string*	Reserved Ports status associated with this VRF.

2.1915 show vrf selection policy

[vrfSelectionPolicies]	*table(string) = TrafficPolicy* see 2.1844.1	VRF selection policy indexed by policy name.

2.1916 show vrf summary

vrfCount	*integer*	Number of VRFs.
vrfUpCount	*integer*	Number of up VRFs.
vrfV4RoutingCount	*integer*	Number of VRFs supporting IPv4 routing.
vrfV6RoutingCount	*integer*	Number of VRFs supporting IPv6 routing.

2.1917 show vrrp [...]

virtualRouters	*list of VrrpRouterDetailBase* see 2.1917.1	Virtual router status.

2.1917.1 VrrpRouterDetailBase

Could also be an instance of:
VrrpRouterDetailV4 2.1917.2
VrrpRouterDetailV6 2.1917.3

description	*string*	Vrrp description.
groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
macAddressInterval	*integer*	The time in seconds between sending neighbor discovery or arp of the virtual mac address.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
masterInterval	*integer*	The master advertisement interval in seconds.
masterPriority	*integer*	Priority of the master virtual router.
preempt	*boolean*	Preemption status.
preemptDelay	*integer*	Preemption delay time in seconds.
preemptReload	*integer*	Preemption reload delay time in seconds.
priority	*integer*	Priority of the virtual router.
skewTime	*floating point number*	The skew of the master down interval in seconds.
state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
trackedObjects **	*list of TrackedObject* see 2.1917.1.1	Vrrp tracked objects. **Since: 4.21.3
version	*integer*	VRRP version.
virtualMac	*MAC address*	The virtual mac address of this vrrp group.
vrIdDisabled **	*boolean*	Virtual router id is disabled. **Since: 4.15.2
vrIdDisabledReason **	*string*	Virtual router id disabled description. **Since: 4.15.2
[vrfName]	*string*	Name of VRF.
vrrpAdvertInterval	*integer*	The time in seconds between sending Vrrp packets.

2.1917.1.1 TrackedObject

interface	*interface name*	Name of the interface.
priority	*integer*	Priority of the tracked object.
shutdown	*boolean*	Shutdown action.

trackedObject	*string*	Tracked object.

2.1917.2 VrrpRouterDetailV4

[authString]	*string*	The authentication string.
[authType]	*enum: md5 | text*	The type of authentication being used.
bfdPeerAddr **	*IPv4 address*	Bfd peer of the virtual router. **Since: 4.21.3
[bfdPeerState] **	*enum: down | shuttingDown | up*	Bfd peer state of the virtual router. **Since: 4.25.2
description	*string*	Vrrp description.
groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
macAddressInterval	*integer*	The time in seconds between sending neighbor discovery or arp of the virtual mac address.
masterAddr	*IPv4 address*	Current address of the master virtual router.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
masterInterval	*integer*	The master advertisement interval in seconds.
masterPriority	*integer*	Priority of the master virtual router.
preempt	*boolean*	Preemption status.
preemptDelay	*integer*	Preemption delay time in seconds.
preemptReload	*integer*	Preemption reload delay time in seconds.
priority	*integer*	Priority of the virtual router.
skewTime	*floating point number*	The skew of the master down interval in seconds.
state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
trackedObjects **	*list of TrackedObject* see 2.1917.1.1	Vrrp tracked objects. **Since: 4.21.3
version	*integer*	VRRP version.
virtualIp	*IPv4 address*	Configured primary virtual Ipv4 address.
virtualIpSecondary	*list of IPv4 address*	Configured secondary virtual Ipv4 addresses.
virtualMac	*MAC address*	The virtual mac address of this vrrp group.
vrIdDisabled **	*boolean*	Virtual router id is disabled. **Since: 4.15.2
vrIdDisabledReason **	*string*	Virtual router id disabled description. **Since: 4.15.2
[vrfName]	*string*	Name of VRF.
vrrpAdvertInterval	*integer*	The time in seconds between sending Vrrp packets.

2.1917.3 VrrpRouterDetailV6

bfdPeerAddr **	*IPv6 address*	Bfd peer of the virtual router. **Since: 4.21.3
[bfdPeerState] **	*enum: down | shuttingDown | up*	Bfd peer state of the virtual router. **Since: 4.25.2
description	*string*	Vrrp description.
groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
linkLocalAddr **	*IPv6 address*	Derived link-local Ipv6 from the VRRP mac or the configured link-local Ipv6 address. **Since: 4.18.0
macAddressInterval	*integer*	The time in seconds between sending neighbor discovery or arp of the virtual mac address.
masterAddr	*IPv6 address*	Current address of the master virtual router.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
masterInterval	*integer*	The master advertisement interval in seconds.
masterPriority	*integer*	Priority of the master virtual router.
preempt	*boolean*	Preemption status.
preemptDelay	*integer*	Preemption delay time in seconds.
preemptReload	*integer*	Preemption reload delay time in seconds.
priority	*integer*	Priority of the virtual router.
skewTime	*floating point number*	The skew of the master down interval in seconds.
state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
trackedObjects **	*list of TrackedObject* see 2.1917.1.1	Vrrp tracked objects. **Since: 4.21.3
version	*integer*	VRRP version.
virtualIpv6	*list of IPv6 address*	Configured secondary virtual Ipv6.
virtualMac	*MAC address*	The virtual mac address of this vrrp group.
vrIdDisabled **	*boolean*	Virtual router id is disabled. **Since: 4.15.2
vrIdDisabledReason **	*string*	Virtual router id disabled description. **Since: 4.15.2
[vrfName]	*string*	Name of VRF.
vrrpAdvertInterval	*integer*	The time in seconds between sending Vrrp packets.

2.1918 show vrrp brief [...]

virtualRouters	*list of VrrpRouterBriefBase* see 2.1918.1	Virtual router status.

2.1918.1 VrrpRouterBriefBase

Could also be an instance of:
VrrpRouterBriefV4 2.1918.2
VrrpRouterBriefV6 2.1918.3

groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
priority	*integer*	Priority of the virtual router.
state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
version	*integer*	VRRP version.
[vrfName]	*string*	Name of VRF.

2.1918.2 VrrpRouterBriefV4

groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
priority	*integer*	Priority of the virtual router.
state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
version	*integer*	VRRP version.
virtualIp	*IPv4 address*	The primary virtual Ipv4 address.
[vrfName]	*string*	Name of VRF.

2.1918.3 VrrpRouterBriefV6

groupId	*integer*	VRRP group ID.
interface	*interface name*	Name of the interface.
masterDownInterval	*integer*	The master down interval of the virtual router in milliseconds.
priority	*integer*	Priority of the virtual router.

state	*enum: backup | stopped | master*	stopped – the virtual router has been shutdown, backup – the virtual router is in backup mode, master – the virtual router is the master of the VRRP group.
stateTransitionTime **	*floating point number*	Timestamp in UTC of last VRRP state transition. **Since: 4.22.0
version	*integer*	VRRP version.
virtualIps	*list of IPv6 address*	Virtual Ipv6 addresses.
[vrfName]	*string*	Name of VRF.

2.1919 show vxlan address-table [...]

revision 2

addresses	*list of VxlanMacAddrListing* see 2.1919.1	List of VXLAN MAC addresses.

2.1919.1 VxlanMacAddrListing

addrType	*enum: dynamic | received | unknown | static | evpn*	Type of the MAC address.
interface	*string*	VXLAN interface for this address.
lastMove	*floating point number*	Time of the last move for this address. A value of 0 indicates the host has not moved.
macAddress	*MAC address*	MAC Address.
moves	*integer*	Number of times this address has moved.
vlanId	*integer*	VLAN Id of MAC address.
vteps	*list of IP address*	List of VTEPs corresponding to this address.

2.1920 show vxlan address-table count [...]

vtepCounts	*table(IPv4 or IPv6 address) = integer*	Count of MAC addresses per VTEP.

2.1921 show vxlan config-sanity [...]

categories	*table(string) = ConfigCheckCategory* see 2.1921.1	Map of category names to check categories.

2.1921.1 ConfigCheckCategory

allCheckPass	*boolean*	Category check passes when all items belonging to the category pass their checks.
description	*string*	Description of category.
detail **	*string*	Detail about this item. **Since: 4.26.2

hasWarning	*boolean*	Category contains warning.
items	*list of ConfigCheckItem* see 2.1921.1.1	List of items in this category.

2.1921.1.1 ConfigCheckItem

checkPass	*boolean*	Item check passed.
detail	*string*	Detail about this item.
hasWarning	*boolean*	Item contains warning.
name	*string*	Name of item.

2.1922 show vxlan control-plane [...]

vlans	*table(numeric string) = VlanControlPlaneConfigModel* see 2.1922.1	VLANs configured for VXLAN, keyed by VLAN.

2.1922.1 VlanControlPlaneConfigModel

controlPlanes	*table(string) = ControlPlaneConfigSourceModel* see 2.1922.1.1	Control planes active in a VLAN, keyed by control plane name.

2.1922.1.1 ControlPlaneConfigSourceModel

sources	*table(string) = ControlPlaneConfigDirectionModel* see 2.1922.1.1.1	Sources of control plane configuration, keyed by source name.

2.1922.1.1.1 ControlPlaneConfigDirectionModel

direction	*enum: both | export | import*	Direction of data sharing for a control plane.

2.1923 show vxlan controller address-table advertised [...]

revision 2

tableType	*string*	Advertised or Received.
vniStatus	*list of VxlanVniStatus* see 2.1711.1	Vxlan tables.

2.1924 show vxlan controller address-table received [...]

revision 2

see 2.1923

2.1925 show vxlan controller arp advertised [...]

protocol **	*enum: arp	nd*	Type of address protocol. **Since: 4.29.2
tableType	*enum: advertised	received*	Type of ARP or ND entries contained in this table.
vlanToArpTable	*table(numeric string) = VxlanArpTable* see 2.1713.1	A mapping from VLAN ID to ARP or ND table.

2.1926 show vxlan controller arp received [...]

see 2.1925

2.1927 show vxlan controller nd advertised [...]

see 2.1925

2.1928 show vxlan controller nd received [...]

see 2.1925

2.1929 show vxlan controller status

[purgeInProgress]	*boolean*	True if purging of HW MAC table entries received from CVX is in progress after CVX restart.
[resyncInProgress]	*boolean*	True if resync with CVX is in progress after CVX restart.
status	*enum: connecting	negotiating	disconnecting	established*	Connection status of Vxlan Control Service with CVX.

2.1930 show vxlan counters learn-restrict COUNTER [...]

counters	*enum: all	prefix	brief*	Select set of per-vlan counters.
learnCountersMap	*table(numeric string) = VxlanPrefixStatusModel* see 2.1930.1	Per VLAN learn list.

2.1930.1 VxlanPrefixStatusModel

learnFrom	*enum: learnFromAny	learnFromList	learnFromFloodList	learnFromDefault*	VTEP prefixes from which we learn.
numMatchAny	*integer*	Number of VTEPs that matched with learnFromAny.

numMatchFloodList	*integer*	Number of VTEPs that matched learnFromFloodList.
numMatchList	*integer*	Number of VTEPs that matched learnFromList.
numRejectFloodList	*integer*	Number of VTEPs dropped due to no match for learnFromFloodList.
numRejectList	*integer*	Number of VTEPs dropped due to no match for learnFromList.
prefixList	*table(IPv4 or IPv6 address prefix) = integer*	Per vlan learn list with number of VTEPs accepted.

2.1931 show vxlan counters software

revision 3

arpPkts	*integer*	Number of ARP packets sent to ARP resolver.
arpReplyL2RefreshNoHer **	*integer*	Number of ARP reply refresh pkts not headend replicated. **Since: 4.27.0
arpReplyRelayNoHer	*integer*	Number of ARP reply packets that were not headend replicated due to ARP relay.
arpReqNoHER	*integer*	Number of ARP request packets that were not headend replicated.
arpRxqRcvd	*integer*	Number of ARP request packets received.
arpRxqRcvdFromVarpMac	*integer*	Number of ARP request packets received from a virtual mac.
arpSendErr	*integer*	Number of send error for ARP packets.
arpSuppressCacheHitMacMiss **	*integer*	Number of ARP cache hits skipped during suppression due to target MAC being not installed. **Since: 4.27.2
arpSuppressSkipDynamicArp	*integer*	Number of dynamic ARP cache hits skipped during suppression due to sender virtual MAC.
arpSuppressSkipLocalIp	*integer*	Number of ARP cache hits skipped during suppression due to target IP address being local.
arpSuppressSkipLocalMac	*integer*	Number of ARP cache hits skipped during suppression due to target being local mac.
arpSuppressSkipMplsMac **	*integer*	Number of ARP cache hits skipped during suppression due to target being behind MPLS tunnel. **Since: 4.27.2
arpSuppressSkipProbe	*integer*	Number of ARP probes skipped ARP suppression.
arpSuppressSkipUnknownMac	*integer*	Number of ARP cache hits skipped during suppression due to sender address being unknown unicast.
arpSupressCache1Miss	*integer*	Number of Arp initial cache misses during Arp suppression.
arpSupressCacheMiss	*integer*	Number of Arp cache misses during Arp suppression.
arpSupressDeny	*integer*	Number of Arp supression packets denied due to ACL.
arpSupressTx	*integer*	Number of ARP cache hits during Arp suppression.
arpSupressTxBcast	*integer*	Number of ARP reply sent to bcast mac during Arp suppression.
arpSupressTxErr	*integer*	Number of ARP reply errors during Arp suppression.
dynamicMacEventSent **	*integer*	Number of dynamic MAC learned events sent to platform. **Since: 4.27.0

encapBytes	*integer*	Number of bytes sent to known remote vteps.
encapDiscardLocalMac	*integer*	Number of packets discarded due to destination being local.
encapDiscardMlagPeerLink	*integer*	Number of discard packets coming in on a MLAG peer link for encapsulation.
encapDiscardRunt	*integer*	Number of discard packets failing size check for encapsulation.
encapDiscardVlanMap	*integer*	Number of discard packets missing VLAN-2-VNI mapping for encapsulation.
encapDiscardVlanRange	*integer*	Number of discard packets failing VLAN range check for encapsulation.
encapDot1qTunnel	*integer*	Number of packets dot1Q received for encapsulation.
encapHwFwd	*integer*	Number of software packets forwarded to remote VTEPs via HW HER.
encapHwFwdErr	*integer*	Number of software packets failed in forwarding to remote VTEPs via HW HER.
encapNoRemoteVtep	*integer*	Number of discard packets destinated to empty flood list for encapsulation.
encapNoRemoteVtepBfd	*integer*	Number of discard packets destinated to empty BFD flood list for encapsulation.
encapPkts	*integer*	Number of packets sent to known remote vteps.
encapReadErr	*integer*	Number of read error packets received for encapsulation.
encapSendDiscardHigh	*integer*	Number of high priority send error packets for encapsulation.
encapSendDiscardLow	*integer*	Number of low priority send error packets for encapsulation.
encapSendErr	*integer*	Number of send error packets for encapsulation.
encapSockDrop	*integer*	Number of pkts dropped by DMA driver socket.
encapTimeout	*integer*	Number of times the sender activity loop yield.
encapTxAnyRemoteVtep	*integer*	Number of BUM packets sent to first vtep on flood list.
encapTxTotalBytes	*integer*	Number of bytes sent after encapsulation.
encapTxTotalIpv6Bytes	*integer*	Total number of bytes encapped with IPv6 underlay.
encapTxTotalIpv6Pkts	*integer*	Total number of packets encapped with IPv6 underlay.
encapTxTotalPkts	*integer*	Number of packets sent after encapsulation.
floodingSuppressCount	*integer*	Number of packets suppressed from getting flooded into Vxlan fabric.
garpPeerSyncMsgCount	*integer*	Number of GRAT ARP sync messages received.
garpPeerSyncMsgErrCount	*integer*	Number of GRAT ARP sync messages received in error.
garpPeerSyncMsgTxCount	*integer*	Number of GRAT ARP sync messges sent.

garpTgtSviSuppress	*integer*	Number of GRAT ARP pkts from SVI suppressed.
installedArpL2Count	*integer*	Number of bridged ARP entries installed.
installedArpL2EntryAgedOut	*integer*	Number of bridged ARP entries removed due to host mac aging.
installedArpL2RefreshSent	*integer*	Number of bridged ARP entry refresh pkts sent.
installedArpL2RefreshSentErr	*integer*	Number of bridged ARP entry refresh failures.
installedArpL2ToL3Count	*integer*	Number of bridged ARP entries migrated from L2 to L3.
installedArpL2ToL3ErrCount	*integer*	Number of bridged ARP entries migrated from L2 to L3 in error.
installedGarpCount	*integer*	Number of GRAT ARP bindings added to ARP.
installedGarpErrCount	*integer*	Number of GRAT ARP bindings added to ARP in error.
installedNaCount	*integer*	Number of dynamic neighbor entries added to neighbor cache.
installedNaErrCount	*integer*	Number of dynamic neighbor entries added to neighbor cache with error.
ipToMacMlagRecd	*integer*	Number of Ip to MAC bindings received from primary Mlag peer.
ipToMacMlagSent	*integer*	Number of Ip to MAC bindings sent to secondary Mlag peer.
naRx	*integer*	Number of Neighbor Advertisement pkts received.
naSuppressSkipInvalidOptPkt	*integer*	Number of Neighbor Advertisements found with invalid or absent options hdr.
naSuppressSkipInvalidPkt	*integer*	Number of Neighbor Advertisements found as invalid.
naSuppressTx	*integer*	Number of Neighbor Advertisement pkts suppressed.
naTgtSvi	*integer*	Number of Neighbor Advertisements not suppressed as target is SVI.
naTgtSviSuppress	*integer*	Number of Neighbor Advertisements suppressed as target is SVI.
noHERCount **	*integer*	Number of frames with no HER requested by dma-driver. **Since: 4.29.2
noIpv4LocalVtepAddr	*integer*	Number of times IPv4 local Vtep address not found while encapsulating the packets.
noIpv4RemoteVtepAddr	*integer*	Number of times IPv4 remote Vtep address not found while encapsulating the packets.
noIpv6LocalVtepAddr	*integer*	Number of times IPv6 local Vtep address not found while encapsulating the packets.
noIpv6RemoteVtepAddr	*integer*	Number of times IPv6 remote Vtep address not found while encapsulating the packets.
nsNASent	*integer*	Number of Neighbor Advertisement proxy pkts sent.
nsReqNoHER	*integer*	ND NS pkts skipped HER since target Ip matched SVI Ip.

nsRx	*integer*	Number of Neighbor Solicitation pkts received.
nsSuppressCache1Miss	*integer*	Number of Neighbor initial cache misses during ND NS proxy.
nsSuppressCacheMiss	*integer*	Number of Neighbor cache misses during ND NS proxy.
nsSuppressDad	*integer*	Number of Neighbor Solicitation DAD pkts suppressed.
nsSuppressDadVarpIp	*integer*	Number of Neighbor Solicitation DAD pkts from kernel to virtual IP suppressed.
nsSuppressDeny	*integer*	Number of ND NS proxy packets denied due to ACL.
nsSuppressNonUnicastSource	*integer*	Number of Neighbor Solicitation pkts from unspecified source.
nsSuppressSkipDynamicArp	*integer*	Number of dynamic Neighbor hits skipped during ND proxy due to source link address being router MAC.
nsSuppressSkipInvalidPkt	*integer*	Number of Neighbor Solicitation pkts found invalid.
nsSuppressSkipLocalIp	*integer*	Number of Neighbor hits skipped during ND NS proxy due to target address being local.
nsSuppressSkipLocalMac	*integer*	Number of Neighbor hits skipped during ND NS proxy due to target link layer being local.
nsSuppressSkipUnknownMac	*integer*	Number of Neighbor hits skipped during ND NS proxy due to sender MAC address being unknown unicast.
nsSuppressTx	*integer*	Number of Neighbor Solicitations suppressed.
nsSuppressTxErr	*integer*	Number of Neighbor Advertisement errors during ND NS proxy.
nsSuppressVirtualLinkLocal	*integer*	Number of Neighbor Solicitation pkts for virtual Link Local IP suppressed.
proxyDisabledBcastArpRx **	*integer*	Total number of broadcast ARPs received from local MAC when ARP proxy is disabled. **Since: 4.27.1
proxyDisabledBcastArpSenderNotLearnedRx **	*integer*	Total number of broadcast ARPs received from unlearned MAC when ARP proxy is disabled. **Since: 4.27.1
proxyDisabledBcastArpSenderRemoteRx **	*integer*	Total number of broadcast ARPs received from remote MAC when ARP proxy is disabled. **Since: 4.27.1
proxyDisabledBcastArpSenderVmacRx **	*integer*	Total number of broadcast ARPs received from virtual MAC when ARP proxy is disabled. **Since: 4.27.1

proxyDisabledMcastNdRx **	*integer*	Total number of multicast NS or NA received from local MAC when ND proxy is disabled. **Since: 4.27.1
proxyDisabledMcastNdSenderNotLearnedRx **	*integer*	Total number of multicast NS/NA received from unlearned MAC when ND proxy is disabled. **Since: 4.27.1
proxyDisabledMcastNdSenderRemoteRx **	*integer*	Total number of multicast NS/NA received from remote MAC when ND proxy is disabled. **Since: 4.27.1
proxyDisabledMcastNdSenderVmacRx **	*integer*	Total number of multicast NS/NA received from virtual MAC when ND proxy is disabled. **Since: 4.27.1
rarpRxCount **	*integer*	Number of RARP pkts. **Since: 4.29.2
rarpRxInvalidCount **	*integer*	Number of RARP pkts that were invalid. **Since: 4.29.2
rsSuppressTx	*integer*	Number of Router Solicitation pkts suppressed.
rxForEncapBytes	*integer*	Number of bytes received for encapsulation.
rxForEncapBytesSockHiPrio	*integer*	Number of bytes received from high priority input queue for encapsulation.
rxForEncapBytesSockLoPrio	*integer*	Number of bytes received from low priority input queue for encapsulation.
rxForEncapPkts	*integer*	Number packets received for encapsulation.
rxForEncapPktsSockHiPrio	*integer*	Number of packets received from high priority input queue for encapsulation.
rxForEncapPktsSockLoPrio	*integer*	Number of packets received from low priority input queue for encapsulation.
vtepToVtepSourcePrunedFrames **	*integer*	Total number of bridged vtep-to-vtep pkts that were source pruned. **Since: 4.27.1

2.1932 show vxlan counters varp

revision 3

arpCacheInstallConflict	*integer*	Number of ARP cache installation conflicts.
arpCacheInstallErr	*integer*	Number of ARP cache installation failure.
arpCacheInstalled	*integer*	Number of ARP packets installed in ARP cache.
arpCacheSyncedInstall-Conflict	*integer*	Number of ARP packets sync install conflicts.
arpCacheSyncedInstallErr	*integer*	Number of ARP packets sync install failures.
arpCacheSyncedRx	*integer*	Number of ARP cache received from MLAG peer.
arpCacheSyncedTx	*integer*	Number of ARP cache sent to MLAG peer.
arpDynamicRemoteArpNotify-SyncedRx **	*integer*	Number of remote ARPs which have been received from MLAG peer. **Since: 4.27.0
arpDynamicRemoteArpNotify-SyncedTx **	*integer*	Number of Remote ARPs which have synced to MLAG peer. **Since: 4.27.0
arpLocalAddressTx	*integer*	Number of ARP request packets rewritten with local IP and bridged.
arpProbeDropForVarpIP **	*integer*	Number of inner ARP probes for VARP IP dropped. **Since: 4.28.0
arpPublishGarpInstall-Conflict **	*integer*	Number of ARP entries failed to install from Gratuitous ARPs due to conflict when ARP publish is enabled. **Since: 4.28.0
arpPublishGarpInstallErr **	*integer*	Number of ARP entries failed to install from Gratuitous ARPs when ARP publish is enabled. **Since: 4.28.0
arpPublishGarpInstalled **	*integer*	Number of ARP entries installed from Gratuitous ARPs when ARP publish is enabled. **Since: 4.28.0
arpPublishGarpRx **	*integer*	Number of Gratuitous ARPs received from a local MAC when ARP publish is enabled. **Since: 4.28.0
arpPublishGarpSenderRemote-Rx **	*integer*	Number of Gratuitous ARPs received from a remote MAC when ARP publish is enabled. **Since: 4.28.0
arpReplyBumListErr	*integer*	Number of received ARP reply packets failed to BUM processing error.
arpReplyHerHwFwd **	*integer*	Number of ARP or IPv6 NA reply pkts sent via H/WHER. **Since: 4.30.1

arpReplyHerHwFwdErr **	*integer*	Number of ARP or IPv6 NA reply pkts sent via H/W HER in error. **Since: 4.30.1
arpReplyHerTx	*integer*	Number of ARP reply packets received for HER encapsulation.
arpReplyMsgToArpMonErr	*integer*	Number of received ARP reply packets failed to ARP monitor processing error.
arpReplyPktSyncedRx	*integer*	Number of ARP reply packets synced over reception.
arpReplyPktSyncedTx	*integer*	Number of ARP reply packets synced over transmission.
arpReplyRcvd	*integer*	Number of ARP reply packets received.
arpReplyRelayTx	*integer*	Number of ARP reply relay packets sent.
arpReplyRewriteDiscard	*integer*	Number of ARP reply packets discard due to missing rewrite pending list match.
arpReplyRx	*integer*	Number of naked ARP reply packets received.
arpReplySrcIpErr	*integer*	Number of received ARP reply packets failed to source IP error check.
arpReplyToBridgeMac-Replicated	*integer*	Number of ARP reply packets rewritten and replicated.
arpReplyToVarpMacNoHer	*integer*	Number of ARP reply packets were not headend replicated.
arpReplyToVarpMacVarpVtepRx	*integer*	Number of ARP reply packets for VARP mac and VARP vtep.
arpReplyTx	*integer*	Number of ARP reply packets received for known VTEPencapsulation.
arpReplyUnmodSent	*integer*	Number of ARP reply packets sent with original source VARP MAC.
arpReqDecapAndTx	*integer*	Number of VXLAN ARP request decapsulated and sent.
arpReqDecapAndTxErr	*integer*	Number of VXLAN ARP request failure in decapsulated and sent.
arpReqDecapAndTxPortIdErr	*integer*	Number of VXLAN ARP request failure due to unknown CPU port id.
arpRxqRcvd	*integer*	Number of received ARP request packets.
arpRxqRcvdFromVarpMac	*integer*	Number of ARP request packets received from a virtual mac.
dropArpReqForNonSwitchIpTo-VarpVtep	*integer*	Number of ARP request packets for non-switch IP to VARP VTEP IP dropped.
dropArpReqForVarpIpToNon-VarpVtep	*integer*	Number of ARP request packets for virtual IP to local VTEP IP dropped.
dropNdNsReqForNonSwitchIp-ToVarpVtep **	*integer*	Number of IPv6 NS requests for non-switch IP to VARP VTEP IP dropped. **Since: 4.28.0
dropNdNsReqForVarpIpToNon-VarpVtep **	*integer*	Number of IPv6 NS requests for virtual IP to local VTEP IP dropped. **Since: 4.28.0
invalidArpRx	*integer*	Number of received ARP packets failure.
invalidIpChecksumRx	*integer*	Number of received packets failed due to invalid IP checksum.

invalidNaOptRx	*integer*	Number of naked neighbor advertisement packets received with invalid or no option hdr.
invalidNaRx	*integer*	Number of naked neighbor advertisement packets received that were invalid.
invalidNdRx **	*integer*	Number of invalid inner IPv6 NS or NA packets received. **Since: 4.28.0
invalidVxlanHdrRx	*integer*	Number of received packets failed due to invalid VXLAN Header check.
ipv4UnderlayPktsRx	*integer*	VXLAN encapped packets received with IPv4 underlay.
ipv6UnderlayPktsRx	*integer*	VXLAN encapped packets received with IPv6 underlay.
localAAInstalledWithout-ActiveRemote **	*integer*	Number of ActiveLocal hosts installed into ARP cache separately from remote binding. **Since: 4.27.1
naRx	*integer*	Number of naked neighbor advertisement packets received.
naToVarpMacNoHer	*integer*	Number of neighbor advertisement packets that were not headend replicated.
ndDadDropForVarpIP **	*integer*	Number of inner IPv6 DAD for VARP IP dropped. **Since: 4.28.0
ndNaReplyRcvd **	*integer*	Number of inner IPv6 NS replies received. **Since: 4.28.0
ndNaReplyToVarpMacVarpVtep-Rx **	*integer*	Number of IPv6 NA reply packets for VARP mac and VARP vtep received. **Since: 4.28.0
ndNaReplyTx **	*integer*	Number of IPv6 NA reply packets sent for known VTEP encapsulation. **Since: 4.28.0
ndNsDecapAndTx **	*integer*	Number of VXLAN NS decapsulated and sent. **Since: 4.30.0
ndNsDecapAndTxErr **	*integer*	Number of VXLAN NS failure in decapsulated and sent. **Since: 4.30.0
ndNsRxqRcvd **	*integer*	Number of inner IPv6 NS requests received. **Since: 4.28.0
ndNsRxqRcvdFromVarpMac **	*integer*	Number of inner IPv6 NS requests received from a virtual mac. **Since: 4.28.0
neighCacheInstallConflict	*integer*	Number of IPv6 neighbor cache installation conflicts.
neighCacheInstallErr	*integer*	Number of IPv6 neighbor cache installation failures.
neighCacheInstalled	*integer*	Number of neighbor entries installed in IPv6 neighbor cache.
neighCacheSyncedInstall-Conflict	*integer*	Number of neighbor entries sync install conflicts.

neighCacheSyncedInstallErr	*integer*	Number of neighbor entries sync install failures.
neighCacheSyncedRx	*integer*	Number of neighbor entries received from MLAG peer.
neighCacheSyncedTx	*integer*	Number of neighbor entries sent to MLAG peer.
neighSyncMsgErrCount	*integer*	Number of neighbor sync messages received in error.
noVlanToVniMapErr	*integer*	Number of received packets failed due to missing VLAN-2-VNI map.
pktTooShortRx	*integer*	Number of received packets failed minimum frame size.
prependDecapForFabric	*integer*	Number of packets decapped and sent on fabric interface from the CPU.
prependDecapForTxFwd	*integer*	Number of packets decapped and sent on txfwd interface from the CPU.
prependDecapForTxRaw	*integer*	Number of packets decapped and sent on txraw interface from the CPU.
unexpectedArpReplyRx	*integer*	Number of received ARP reply packets failure.
unexpectedEtherTypeRx	*integer*	Number of received packets failed due to unexpected ethertype.
unexpectedInnerNotArpRx	*integer*	Number of received packets failed due to inner ARP or IPv6 ND frames missing.
unexpectedIpProtoRx	*integer*	Number of received packets failed due to unexpected IP protocol value.
unexpectedNonIpRx	*integer*	Number of received packets failed due to non IP packet.
unexpectedUdpPortRx	*integer*	Number of received packets failed due to unexpected UDP destination port number.
unexpectedVlanIdRx **	*integer*	Number of received packets failed due to unexpected VLAN range. **Since: 4.27.0
vxlanEncapArpRx	*integer*	Number of VXLAN ARP packets received.
vxlanEncapNdRx **	*integer*	Number of inner IPv6 NS or NA packets received. **Since: 4.28.0
vxlanEvpnPeerFilterDecapped **	*integer*	Number of VXLAN ARP packets filtered from multihoming peer and decapped. **Since: 4.27.1
vxlanSecArpReplyTx **	*integer*	Number of VxlanSec ARP replies sent. **Since: 4.27.1
vxlanSecArpReplyTxErr **	*integer*	Number of VxlanSec ARP replies sent in error. **Since: 4.27.1
vxlanSecArpReqForVarpIpDrop **	*integer*	Number of VxlanSec ARP requests to VARP IP dropped. **Since: 4.27.1
vxlanSecEncapArpRx **	*integer*	Number of VxlanSec ARP packets received. **Since: 4.27.1

vxlanSecEncapNdRx **	*integer*	Number of VxlanSec inner IPv6 NS or NA packets received. **Since: 4.28.0
vxlanSecNdNsForVarpIpDrop **	*integer*	Number of VxlanSec inner IPv6 NS requests to VARP IP dropped. **Since: 4.28.0
vxlanSecNdReplyTx **	*integer*	Number of VxlanSec inner IPv6 NA reply sent. **Since: 4.28.0
vxlanSecNdReplyTxErr **	*integer*	Number of VxlanSec inner IPv6 NA reply sent in error. **Since: 4.28.0

2.1933 show vxlan counters vni [...]

vniInDottedNotation	*boolean*	True if VNI string is in dotted notation.
vnis	*table(string) = VxlanVniCounters* see 2.1933.1	Mapping between VNI and its counters.

2.1933.1 VxlanVniCounters

[decapBUMPkts]	*integer*	Layer2 BUM (Broadcast, Unknown, and Multicast) packets decapped for a VNI.
[decapBytes]	*integer*	Bytes decapped for a VNI.
[decapDropExcptPkts]	*integer*	Dropped and Exception packets decapped for a VNI.
[decapKnownUcastPkts]	*integer*	Layer2 Known Unicast packets decapped for a VNI.
[encapBUMPkts]	*integer*	Layer2 BUM (Broadcast, Unknown, and Multicast) packets for a VNI before replication for encap.
[encapBytes]	*integer*	Bytes encapped for a VNI.
[encapDropPkts]	*integer*	Dropped and Exception packets encapped for a VNI.
[encapPkts]	*integer*	Packets encapped for a VNI.

2.1934 show vxlan counters vtep [...]

[unlearntVteps]	*VxlanVtepCounters* see 2.1934.1	Decap Counters for unlearnt VTEPs.
vteps	*table(IPv4 or IPv6 address) = VxlanVtepCounters* see 2.1934.1	Mapping between IP address and VTEP counters.

2.1934.1 VxlanVtepCounters

[decapBUMPkts]	*integer*	Layer2 BUM (Broadcast, Unknown, and Multicast) packets decapped from VTEP.
[decapBytes]	*integer*	Bytes decapped from VTEP.

[decapDropExcptPkts]	*integer*	Dropped and Exception packets decapped from VTEP.
[decapKnownUcastPkts]	*integer*	Layer2 Known Unicast packets decapped from VTEP.
[decapPkts] **	*integer*	Packets decapped from VTEP. **Since: 4.22.1
[encapBumPkts] **	*integer*	BUM Packets encapped to VTEP. **Since: 4.22.1
[encapBytes]	*integer*	Bytes encapped to VTEP.
[encapDropExcptPkts]	*integer*	Dropped and Exception packets encapped to VTEP.
[encapPkts]	*integer*	Packets encapped to VTEP.

2.1935 show vxlan flood traffic

arpFlooding	*boolean*	True if flooding of ARP traffic is enabled.
defaultFlooding	*boolean*	True if default flooding of all traffic is enabled.
ndFlooding	*boolean*	True if flooding of ND traffic is enabled.
unknownMacFlooding	*boolean*	True if flooding of unknown MAC traffic is enabled.

2.1936 show vxlan flood vtep [...]

floodMap	*table(string) = VxlanVtepListModel* see 2.1936.1	Per VLAN flood list.

2.1936.1 VxlanVtepListModel

vteps	*list of IP address*	VTEP IP address.

2.1937 show vxlan learn-restrict vtep [...]

learnMap	*table(numeric string) = VxlanPrefixListModel* see 2.1937.1	Per vlan learn list.

2.1937.1 VxlanPrefixListModel

learnFrom	*enum: learnFromAny | learnFromList | learnFromFloodList | learnFromDefault*	VTEP prefixes from which we learn.
prefixList	*list of IP address prefix*	VTEP IP address (prefix).

2.1938 show vxlan policers vni [...]

vxlanConfig	*table(string) = VxlanConfigModel* see 2.1938.1	A mapping of VXLAN interface to its configuration.

2.1938.1 VxlanConfigModel

vniQosInfo	*table(numeric string) = VxlanVniQosInfoModel* see 2.1938.1.1	A mapping of VNI to its policer information.

2.1938.1.1 VxlanVniQosInfoModel

[extVlan]	*integer*	VLAN-VNI-VRF mapping.
[iPolicerInfo] **	*VxlanVniPolicer* see 2.1938.1.1.1	Ingress policer information. **Since: 4.29.0
[oPolicerInfo] **	*VxlanVniPolicer* see 2.1938.1.1.1	Egress policer information. **Since: 4.29.0

2.1938.1.1.1 VxlanVniPolicer

mode	*VxlanVniPolicerModeEnum* see 2.1938.1.1.1.2	Policer mode configuration.
name	*string*	Policer name.
status	*VxlanVniPolicerHwStateEnum* see 2.1938.1.1.1.1	Policer hardware programming status.

2.1938.1.1.1.1 VxlanVniPolicerHwStateEnum

hwState	*enum: inactive | active*	Policer hardware state.

2.1938.1.1.1.2 VxlanVniPolicerModeEnum

value	*enum: profile | group*	Policer mode.

2.1939 show vxlan policers vni counters [...]

egrInterfaces	*table(Interface name) = VxlanVniPolicersAllCountersInfoModel* see 2.1939.1	Per VXLAN interface collection for egress policers.
ingInterfaces	*table(Interface name) = VxlanVniPolicersAllCountersInfoModel* see 2.1939.1	Per VXLAN interface collection for ingress policers.

2.1939.1 VxlanVniPolicersAllCountersInfoModel

vniPolicers	*table(numeric string) = VxlanVniPolicersCounters-InfoModel* see 2.1939.1.1	Per VNI policer collection.

2.1939.1.1 VxlanVniPolicersCountersInfoModel

counters	*VniPolicingCounters* see 2.1939.1.1.1	VNI Policer Counters.
mode	*VxlanVniPolicerModeEnum* see 2.1938.1.1.1.2	Policer mode configuration.
[policerName]	*string*	Policer name.

2.1939.1.1.1 VniPolicingCounters

conformedBytes	*integer*	Conformed number of bytes to policer.
conformedPackets	*integer*	Conformed number of packets to policer.
exceededBytes	*integer*	Exceeded number of bytes to policer.
exceededPackets	*integer*	Exceeded number of packets to policer.

2.1940 show vxlan qos

revision 2

[decapDscpEcnPropagation-Disabled]	*boolean*	VXLAN DSCP and ECN propagation is disabled for VXLAN decapsulated packets.
[dscpEcnRewriteBridged-Enabled] **	*boolean*	VXLAN DSCP and ECN rewrite is enabled for VXLAN bridged packets. **Since: 4.30.1
[ecnPropagation]	*boolean*	VXLAN ECN propagation is enabled.
outerDscpFromInnerDscp	*boolean*	Outer DSCP is derived from Inner DSCP.
routingDscpRewriteSupported	*boolean*	DSCP rewrite is supported on VXLAN routed packets.
trafficClassFromOuterDscp	*boolean*	Traffic Class is derived from Outer DSCP.

2.1941 show vxlan security profile

vteps	*table(IPv4 or IPv6 address) = string*	A mapping of VXLAN VTEP to its security profile.

2.1942 show vxlan vni [...]

vxlanIntfs	*table(Interface name) = VxlanVniVlanCollection* see 2.1942.1	A mapping of VXLAN interface to VNI collection.

2.1942.1 VxlanVniVlanCollection

vniBindings	*table(numeric string) = VxlanVniVlanMapping-Model* see 2.1942.1.1	VNI to VLAN mapping.
vniBindingsToVrf **	*table(numeric string) = VxlanVniVrfMappingModel* see 2.1942.1.2	VNI to VRF mapping. **Since: 4.24.2

2.1942.1.1 VxlanVniVlanMappingModel

[dynamicVlan]	*boolean*	VLAN is dynamically configured.
interfaces	*table(Interface name) = VxlanIntfDot1qModel* see 2.1942.1.1.1	Ports associated with a VNI.
source	*string*	Source of VNI to VLAN mapping, (VCS, DYNVTEP, STATIC).
vlan	*integer*	VLAN ID.

2.1942.1.1.1 VxlanIntfDot1qModel

dot1q	*integer*	Dot1q tagged VLAN.
[dot1qTunnel] **	*boolean*	Dot1q-tunneled VLAN. **Since: 4.21.3
[error]	*string*	Error with VLAN translation.

2.1942.1.2 VxlanVniVrfMappingModel

source	*string*	Source of VNI to dynamic VLAN mapping.
vlan	*integer*	Dynamic VLAN ID.
[vrfName]	*string*	VRF which is associated with the VNI.

2.1943 show vxlan vni summary

vxlanVniSources	*table(string) = integer*	A mapping of VXLAN source to source count.

2.1944 show vxlan vtep [...]

interfaces	*table(Interface name) = VxlanInterface* see 2.1944.3	Mapping from VXLAN interface to list of VTEP addresses.
vtepTunnelTypes **	*table(IPv4 or IPv6 address) = TunnelTypeList* see 2.1944.2	Mapping of VTEP to list of tunnel types. **Since: 4.25.0
[vteps] **	*table(IPv4 or IPv6 address) = RemoteVtepConfig* see 2.1944.1	A mapping of remote VTEP to details. **Since: 4.24.2

2.1944.1 RemoteVtepConfig

[evpnDomain] **	*enum: remoteDomain | localDomain*	Domain in which VTEP is located. **Since: 4.27.0
learnedVia	*enum: dataPlane | controlPlane*	Source from which remote VTEP was learned.
macAddressLearning	*enum: datapath | controlPlane*	Source for MAC address learning for VTEP.

2.1944.2 TunnelTypeList

tunnelTypes	*list of TunnelType* see 2.1944.2.1	List of tunnel types.

2.1944.2.1 TunnelType

tunnelType	*enum: flood | unicast | ecmp | multicast*	VXLAN tunnel type.

2.1944.3 VxlanInterface

vteps	*list of IP address*	List of VTEPs.

2.1945 show vxlan vtep summary

vxlanInterfaces	*table(Interface name) = integer*	Total VTEPs by interface.

2.1946 show world

[breakable]	*boolean*	Is that world breakable.
countries	*table(string) = Country* see 2.1946.1	Countries of the world.
population	*integer*	Population living in world.

2.1946.1 Country

capital	*string*	Name of country's capital.
[cities]	*table(string) = City* see 2.1946.1.1	Cities of the county.
[elections]	*string*	date of next election.
population	*integer*	Population of this country in millions.

2.1946.1.1 City

mayor	*string*	Name of the city's mayor.
[pools]	*integer*	Number of public pools in city.
population	*integer*	Population of this city in millions.

2.1947 show world bad

see 2.1946

2.1948 show world stream [...]

someInt	*integer*	some help.

2.1949 show xmpp neighbors

neighbors	*table(string) = Neighbor* see 2.1949.1	Collection of XMPP neighbor details keyed by neighbor base JID.

2.1949.1 Neighbor

lastUpdateTimestamp	*floating point number*	Last time the neighbor status changed in seconds since epoch, UTC.
state	*enum: notPresent | present | unknown*	The presence state of this neighbor learnt from XMPP.
statusMessage	*string*	The XMPP status message of this neighbor.

2.1950 show xmpp status

connectionState	*enum: authenticationFailed | connected | notConnected | unknown | encryptionFailed*	XMPP server connection status.
domainName	*string*	Default domain name appended to bare usernames.
enabled	*boolean*	Whether the XMPP agent is enabled.
initialPrivLevel	*integer*	The initial privilege level for command authorization when not using AAA.
port	*integer*	The TCP port number used for XMPP connections.
serverAddress	*string*	The IP address or hostname of the XMPP server.
srcInterface **	*interface name*	Interface providing source IP addressfor XMPP. **Since: 4.20.0
srcIpAddr **	*IPv4 address*	Source ip address of Xmpp packets. **Since: 4.20.0
statusMessage	*string*	The XMPP agent's current status message.
username	*string*	The XMPP base jid (email address) for the XMPP server.
vrfName	*string*	The current VRF name the agent is running in.

2.1951 show xmpp switch-group

`groups`	*table(string) = Group* see 2.1951.1	Collection of group details keyed by switch group name (aka XMPP chat room JIDs).

2.1951.1 Group

`enabled`	*boolean*	Whether we are connected to this room.

2.1952 show zerotouch

`mode`	*enum: disabled | enabled | active*	ZeroTouch Mode.
`scriptTimeout`	*integer*	Script timeout in secs.

2.1953 traceroute mpls generic ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)) [...]

revision 2

traceroute	*MplsTracerouteHdr* see 2.1953.1	Traceroute request.

2.1953.1 MplsTracerouteHdr

dsInfoType	*enum: dsmap | ddmap*	Downstream information type.
errInfo	*list of ErrInfo* see 2.2.1.2	Errors.
[ldpModel]	*GenericModel* see 2.2.1.3	MPLS LDP traceroute model.
protocol	*enum: segment-routing-ospf | nexthop-group-tunnel | rsvp | segment-routing | SrTePolicy | mldp | bgpLu | static | ldp | vpn | raw | generic | nexthop-group | pwLdp*	Protocol.
[rsvpModel] **	*RsvpModel* see 2.2.1.6	MPLS RSVP traceroute model. **Since: 4.30.0
[srModel] **	*SrModel* see 2.2.1.7	MPLS SR traceroute model. **Since: 4.28.2
[statsModel] **	*list of MplsVia* see 2.1953.1.1	Statistics summary per Via. **Since: 4.30.1
viaModel	*list of MplsVia* see 2.1953.1.1	Vias.

2.1953.1.1 MplsVia

[interface] **	*string*	Interface name. **Since: 4.30.1
labelStack	*list of integer*	LabelStack.
nextHopIp	*string*	NextHop IP.
path	*list of MplsPath* see 2.1953.1.1.1	Via paths.
[stats] **	*MplsTracerouteStatistics-Summary* see 2.1953.1.1.2	MPLS traceroute statistics summary. **Since: 4.30.1

2.1953.1.1.1 MplsPath

hopPkt	*list of HopPktInfo* see 2.1953.1.1.1.1	HopPktInfos.
pathResult	*enum: Successfully explored path. | Error: Could not probe further...*	Result of path exploration.

2.1953.1.1.1.1 HopPktInfo

dsInfo	*list of DownstreamInfo* see 2.1953.1.1.1.2	Downstream infos.
hopMtu	*integer*	Hop MTU.
[intfAndLabelStackInfo]	*IntfAndLabelStackInfo* see 2.1953.1.1.1.4	Intf and labelStack info.
oneWayDelay **	*integer*	One way delay (us). **Since: 4.29.1
replyHost	*string*	Reply address.
retCode	*enum: noRetCode* | *seeDdmTlv* | *repRouterEgress* | *usIntfUnknown* | *noMappingForFec* | *labelSwitchedAtStackDep* | *labelSwitchedNoMplsFwd* | *malFormEchoRequest* | *reservedRetCode* | *labelMappingMismatch* | *protocolIntfMismatch* | *lblSwitchedFecChange* | *premTermination* | *noLabelEntryAtStackDep* | *dsMappingMismatch* | *tlvNotUnderstood*	Return code.
roundTrip	*integer*	Roundtrip time (ms).
roundTripUs **	*integer*	RoundTrip time (us). **Since: 4.29.1
ttl	*integer*	Time to live.

2.1953.1.1.1.2 DownstreamInfo

addrType	*enum: ipv4Unnumbered* | *ipv6Numbered* | *ipv6Unnumbered* | *ipv4Numbered*	Address type.
dsIntfAddr	*string*	Downstream interface address.
dsIntfIndex	*integer*	Downstream interface index.
dsIpAddr	*string*	Downstream IP address.
dsType	*enum: downstreamMap* | *downstreamDetailedMap*	Downstream information type.
labelStack	*list of string*	Downstream label stack.
[popFecTypeList] **	*FecTypeList* see 2.1953.1.1.1.3	List of FEC types to pop. **Since: 4.30.1
[pushFecTypeList] **	*FecTypeList* see 2.1953.1.1.1.3	List of FEC types to push. **Since: 4.30.1

retCode	*enum: noRetCode* | *seeDdmTlv* | *repRouterEgress* | *usIntfUnknown* | *noMappingForFec* | *labelSwitchedAtStackDep* | *labelSwitchedNoMplsFwd* | *malFormEchoRequest* | *reservedRetCode* | *labelMappingMismatch* | *protocolIntfMismatch* | *lblSwitchedFecChange* | *premTermination* | *noLabelEntryAtStackDep* | *dsMappingMismatch* | *tlvNotUnderstood*	Return code.

2.1953.1.1.1.3 FecTypeList

fecTypeList	*list of string*	FEC TLV type list.

2.1953.1.1.1.4 IntfAndLabelStackInfo

addr	*string*	IP address.
addrType	*enum: ipv4Unnumbered* | *ipv6Numbered* | *ipv6Unnumbered* | *ipv4Numbered*	Address type.
intfAddr	*string*	Interface address.
intfIndex	*integer*	Interface index.
labelStack	*list of string*	Received label stack.

2.1953.1.1.2 MplsTracerouteStatisticsSummary

[mplsStatistics]	*list of MplsTracerouteStatistics* see 2.1953.1.1.2.1	MPLS traceroute statistics details.
packetLoss	*integer*	Percentage of packets lost.
packetsReceived	*integer*	Packets received.
packetsTransmitted	*integer*	Packets transmitted.
replyTime	*integer*	Reply timestamp time.

2.1953.1.1.2.1 MplsTracerouteStatistics

destination	*string*	Reply host address.
packetsReceived	*integer*	Packets received from destination.
roundTripTimeAvgUs	*integer*	Average round trip time in microseconds.
roundTripTimeMaxUs	*integer*	Maximum round trip time in microseconds.

roundTripTimeMinUs	*integer*	Minimum round trip time in microseconds.

2.1954 traceroute mpls ldp ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_M)) [...]

revision 2

see 2.1953

2.1955 traceroute mpls p2mp mldp ROOTADDR ((lsp-id OPQVAL) | (multicast ipv4 SOURCEADDR GROUPADDR)) [...]

revision 2

see 2.1953

2.1956 traceroute mpls rsvp session ((id SESSION_ID lsp LSP) | (name SESSION_NAME)) [...]

revision 2

see 2.1953

2.1957 traceroute mpls rsvp tunnel TUNNEL_NAME [...]

revision 2

see 2.1953

2.1958 traceroute mpls segment-routing ((((ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (ipv6 IPV6_ADDRESS) | (isis ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK))) | (isis ipv6 IPV6_ADDRESS))) | (ospf ip (IP_ADDRESS | (IP_ADDRESS_ADDR IP_ADDRESS_MASK)))) [...]

revision 2

see 2.1953