IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru

Simple Network Management Protocol Reference Guide

IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru

Simple Network Management Protocol Reference Guide

Note: Before using this information and the product it supports, read the general information in Appendix B, "Notices, on page 519.

First Edition, February 2012

Contents

Chapter 1. Simple Network Management Protocol 1

Related documentation 1
Notices and statements in this document 3

Chapter 2. SNMP overview 5

SNMP interface objectives 5
Manager and agent 5
Traps 6
Management information base 7
User datagram protocol 7
Numbering system conventions 7

Chapter 3. Configuring a switch 9

System Specifications and requirements 9
Configuring a switch using the command line interface 9
Configuring a switch Using QuickTools 13

Chapter 4. MIB-II objects 15

Groups in MIB-II 15
System group 16
sysDescr (1.3.6.1.2.1.1.1) 16
sysObjectID (1.3.6.1.2.1.1.2) 17
sysUpTime (1.3.6.1.2.1.1.3) 18
sysContact (1.3.6.1.2.1.1.4) 19
sysName (1.3.6.1.2.1.1.5) 20
sysLocation (1.3.6.1.2.1.1.6) 21
sysServices (1.3.6.1.2.1.1.7) 22
The Interfaces Group 23
ifNumber (1.3.6.1.2.1.2.1) 23
The interfaces table 24
ifIndex (1.3.6.1.2.1.2.2.1.1) 24
ifDescr (1.3.6.1.2.1.2.2.1.2) 25
ifType (1.3.6.1.2.1.2.2.1.3) 26
ifMtu (1.3.6.1.2.1.2.2.1.4) 27
ifSpeed (1.3.6.1.2.1.2.2.1.5) 28
ifPhysAddress (1.3.6.1.2.1.2.2.1.6) 29
ifAdminStatus (1.3.6.1.2.1.2.2.1.7) 30
ifOperStatus (1.3.6.1.2.1.2.2.1.8) 31
ifLastChange (1.3.6.1.2.1.2.2.1.9) 32
ifInOctets (1.3.6.1.2.1.2.2.1.10) 33
ifInUcastPkts (1.3.6.1.2.1.2.2.1.11) 34
ifInNUcastPkts (1.3.6.1.2.1.2.2.1.12) 35
ifInDiscards (1.3.6.1.2.1.2.2.1.13) 36
ifInErrors (1.3.6.1.2.1.2.2.1.14) 37
ifInUnknownProtos (1.3.6.1.2.1.2.2.1.15) 38
ifOutOctets (1.3.6.1.2.1.2.2.1.16) 39
ifOutUcastPkts (1.3.6.1.2.1.2.2.1.17) 40
ifOutNUcastPkts (1.3.6.1.2.1.2.2.1.18) 41
ifOutDiscards (1.3.6.1.2.1.2.2.1.19) 42
ifOutErrors (1.3.6.1.2.1.2.2.1.20) 43

ifOutQLen (1.3.6.1.2.1.2.2.1.21) . . . 44
ifSpecific (1.3.6.1.2.1.2.2.1.22) . . . 45
The Address Translation Group . . . 46
atIfIndex (1.3.6.1.2.1.3.1.1.1) . . . 46
atPhysAddress (1.3.6.1.2.1.3.1.1.2) . . . 47
atNetAddress (1.3.6.1.2.1.3.1.1.3) . . . 48
The IP Group . . . 49
ipForwarding (1.3.6.1.2.1.4.1) . . . 49
ipDefaultTTL (1.3.6.1.2.1.4.2) . . . 50
ipInReceives (1.3.6.1.2.1.4.3) . . . 51
ipInHdrErrors (1.3.6.1.2.1.4.4) . . . 52
ipInAddrErrors (1.3.6.1.2.1.4.5) . . . 53
ipForwDatagrams (1.3.6.1.2.1.4.6) . . . 54
ipInUnknownProtos (1.3.6.1.2.1.4.7) . . . 55
ipInDiscards (1.3.6.1.2.1.4.8) . . . 56
ipInDelivers (1.3.6.1.2.1.4.9) . . . 57
ipOutRequests (1.3.6.1.2.1.4.10) . . . 58
ipOutDiscards (1.3.6.1.2.1.4.11) . . . 59
ipOutNoRoutes (1.3.6.1.2.1.4.12) . . . 60
ipReasmTimeout (1.3.6.1.2.1.4.13) . . . 61
ipReasmReqds (1.3.6.1.2.1.4.14) . . . 62
ipReasmOKs (1.3.6.1.2.1.4.15) . . . 63
ipReasmFails (1.3.6.1.2.1.4.16) . . . 64
ipFragOKs (1.3.6.1.2.1.4.17) . . . 65
ipFragFails (1.3.6.1.2.1.4.18) . . . 66
ipFragCreates (1.3.6.1.2.1.4.19) . . . 67
The IP address table . . . 68
ipAdEntAddr (1.3.6.1.2.1.4.20.1.1) . . . 68
ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2) . . . 69
ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3) . . . 70
ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4) . . . 71
ipAdEntReasmMaxSize (1.3.6.1.2.1.4.20.1.5) . . . 72
The IP Routing Table . . . 73
ipRouteDest (1.3.6.1.2.1.4.21.1.1) . . . 73
ipRouteIfIndex (1.3.6.1.2.1.4.21.1.2) . . . 74
ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3) . . . 75
ipRouteMetric2 (1.3.6.1.2.1.4.21.1.4) . . . 76
ipRouteMetric3 (1.3.6.1.2.1.4.21.1.5) . . . 77
ipRouteMetric4 (1.3.6.1.2.1.4.21.1.6) . . . 78
ipRouteNextHop (1.3.6.1.2.1.4.21.1.7) . . . 79
ipRouteType (1.3.6.1.2.1.4.21.1.8) . . . 80
ipRouteProto (1.3.6.1.2.1.4.21.1.9) . . . 81
ipRouteAge (1.3.6.1.2.1.4.21.1.10) . . . 82
ipRouteMask (1.3.6.1.2.1.4.21.1.11) . . . 83
ipRouteMetric5 (1.3.6.1.2.1.4.21.1.12) . . . 84
ipRouteInfo (1.3.6.1.2.1.4.21.1.13) . . . 85
The IP Address Translation Table . . . 86
ipNetToMediaIfIndex (1.3.6.1.2.1.4.22.1.1) . . . 86
ipNetToMediaPhysAddress (1.3.6.1.2.1.4.22.1.2) . . . 87
ipNetToMediaNetAddress (1.3.6.1.2.1.4.22.1.3) . . . 88
ipNetToMediaType (1.3.6.1.2.1.4.22.1.4) . . . 89
Additional IP Objects . . . 90
ipRoutingDiscards (1.3.6.1.2.1.4.23) . . . 90
The ICMP group . . . 91
icmpInMsgs (1.3.6.1.2.1.5.1) . . . 91

icmpInErrors (1.3.6.1.2.1.5.2) . . . 92
icmpInDestUnreachs (1.3.6.1.2.1.5.3) . . . 93
icmpInTimeExcds (1.3.6.1.2.1.5.4) . . . 94
icmpInParmProbs (1.3.6.1.2.1.5.5) . . . 95
icmpInSrcQuenchs (1.3.6.1.2.1.5.6) . . . 96
icmpInRedirects (1.3.6.1.2.1.5.7) . . . 97
icmpInEchos (1.3.6.1.2.1.5.8) . . . 98
icmpInEchoReps (1.3.6.1.2.1.5.9) . . . 99
icmpInTimestamps (1.3.6.1.2.1.5.10) . . . 100
icmpInTimestampReps (1.3.6.1.2.1.5.11) . . . 101
icmpInAddrMasks (1.3.6.1.2.1.5.12) . . . 102
icmpInAddrMaskReps (1.3.6.1.2.1.5.13) . . . 103
icmpOutMsgs (1.3.6.1.2.1.5.14) . . . 104
icmpOutErrors (1.3.6.1.2.1.5.15) . . . 105
icmpOutDestUnreachs (1.3.6.1.2.1.5.16) . . . 106
icmpOutTimeExcds (1.3.6.1.2.1.5.17) . . . 107
icmpOutParmProbs (1.3.6.1.2.1.5.18) . . . 108
icmpOutSrcQuenchs (1.3.6.1.2.1.5.19) . . . 109
icmpOutRedirects (1.3.6.1.2.1.5.20) . . . 110
icmpOutEchos (1.3.6.1.2.1.5.21) . . . 111
icmpOutEchoReps (1.3.6.1.2.1.5.22) . . . 112
icmpOutTimestamps (1.3.6.1.2.1.5.23) . . . 113
icmpOutTimestampReps (1.3.6.1.2.1.5.24) . . . 114
icmpOutAddrMasks (1.3.6.1.2.1.5.25) . . . 115
icmpOutAddrMaskReps (1.3.6.1.2.1.5.26) . . . 116
The TCP group . . . 117
tcpRtoAlgorithm (1.3.6.1.2.1.6.1) . . . 117
tcpRtoMin (1.3.6.1.2.1.6.2) . . . 118
tcpRtoMax (1.3.6.1.2.1.6.3) . . . 119
tcpMaxConn (1.3.6.1.2.1.6.4) . . . 120
tcpActiveOpens (1.3.6.1.2.1.6.5) . . . 121
tcpPassiveOpens (1.3.6.1.2.1.6.6) . . . 122
tcpAttemptFails (1.3.6.1.2.1.6.7) . . . 123
tcpEstabResets (1.3.6.1.2.1.6.8) . . . 124
tcpCurrEstab (1.3.6.1.2.1.6.9) . . . 125
tcpInSegs (1.3.6.1.2.1.6.10) . . . 126
tcpOutSegs (1.3.6.1.2.1.6.11) . . . 127
tcpRetransSegs (1.3.6.1.2.1.6.12) . . . 128
The TCP connection table . . . 129
tcpConnState (1.3.6.1.2.1.6.13.1.1) . . . 129
tcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2) . . . 130
tcpConnLocalPort (1.3.6.1.2.1.6.13.1.3) . . . 131
tcpConnRemAddress (1.3.6.1.2.1.6.13.1.4) . . . 132
tcpConnRemPort (1.3.6.1.2.1.6.13.1.5) . . . 133
Additional TCP objects . . . 134
tcpInErrs (1.3.6.1.2.1.6.14) . . . 134
tcpOutRsts (1.3.6.1.2.1.6.15) . . . 135
The UDP group . . . 136
udpInDatagrams (1.3.6.1.2.1.7.1) . . . 136
udpNoPorts (1.3.6.1.2.1.7.2) . . . 137
udpInErrors (1.3.6.1.2.1.7.3) . . . 138
udpOutDatagrams (1.3.6.1.2.1.7.4) . . . 139
The UDP listener table . . . 140
udpLocalAddress (1.3.6.1.2.1.7.5.1.1) . . . 140
udpLocalPort (1.3.6.1.2.1.7.5.1.2) . . . 141

The EGP Group . . . 142
egpInMsgs (1.3.6.1.2.1.8.1) . . . 142
egpInErrors (1.3.6.1.2.1.8.2) . . . 143
egpOutMsgs (1.3.6.1.2.1.8.3) . . . 144
egpOutErrors (1.3.6.1.2.1.8.4) . . . 145
The EGP neighbor table . . . 146
egpNeighState (1.3.6.1.2.1.8.5.1.1) . . . 146
egpNeighAddr (1.3.6.1.2.1.8.5.1.2) . . . 147
egpNeighAs (1.3.6.1.2.1.8.5.1.3) . . . 148
egpNeighInMsgs (1.3.6.1.2.1.8.5.1.4) . . . 149
egpNeighInErrs (1.3.6.1.2.1.8.5.1.5) . . . 150
egpNeighOutMsgs (1.3.6.1.2.1.8.5.1.6) . . . 151
egpNeighOutErrs (1.3.6.1.2.1.8.5.1.7) . . . 152
egpNeighInErrMsgs (1.3.6.1.2.1.8.5.1.8) . . . 153
egpNeighOutErrMsgs (1.3.6.1.2.1.8.5.1.9) . . . 154
egpNeighStateUps (1.3.6.1.2.1.8.5.1.10) . . . 155
egpNeighStateDowns (1.3.6.1.2.1.8.5.1.11) . . . 156
egpNeighIntervalHello (1.3.6.1.2.1.8.5.1.12) . . . 157
egpNeighIntervalPoll (1.3.6.1.2.1.8.5.1.13) . . . 158
egpNeighMode (1.3.6.1.2.1.8.5.1.14) . . . 159
egpNeighEventTrigger (1.3.6.1.2.1.8.5.1.15) . . . 160
egpAs (1.3.6.1.2.1.8.6) . . . 161
The transmission group . . . 162
The dot3StatTable . . . 162
EtherLike-MIB:dot3StatsIndex (1.3.6.1.2.1.10.7.2.1.1) . . . 162
EtherLike-MIB:dot3StatsFCSErrors (1.3.6.1.2.1.10.7.2.1.3) . . . 163
EtherLike-MIB:dot3StatsInternalMacReceiveErrors (1.3.6.1.2.1.10.7.2.1.16) . . . 164
EtherLike-MIB:dot3StatsSymbolErrors (1.3.6.1.2.1.10.7.2.1.18) . . . 165
The dot3ControlTable . . . 166
EtherLike-MIB:dot3ControlFunctionsSupported (1.3.6.1.2.1.10.7.9.1.1) . . . 166
The dot3PauseTable . . . 167
EtherLike-MIB:dot3PauseAdminMode (1.3.6.1.2.1.10.7.10.1.1) . . . 167
EtherLike-MIB:dot3PauseOperMode (1.3.6.1.2.1.10.7.10.1.2) . . . 168
EtherLike-MIB:dot3InPauseFrames (1.3.6.1.2.1.10.7.10.1.3) . . . 169
EtherLike-MIB:dot3OutPauseFrames (1.3.6.1.2.1.10.7.10.1.4) . . . 170
The SNMP group . . . 171
snmpInPkts (1.3.6.1.2.1.11.1) . . . 171
snmpOutPkts (1.3.6.1.2.1.11.2) . . . 172
snmpInBadVersions (1.3.6.1.2.1.11.3) . . . 173
snmpInBadCommunityNames (1.3.6.1.2.1.11.4) . . . 174
snmpInBadCommunityUses (1.3.6.1.2.1.11.5) . . . 175
snmpInASNParseErrs (1.3.6.1.2.1.11.6) . . . 176
snmpInTooBigs (1.3.6.1.2.1.11.8) . . . 177
snmpInNoSuchNames (1.3.6.1.2.1.11.9) . . . 178
snmpInBadValues (1.3.6.1.2.1.11.10) . . . 179
snmpInReadOnlys (1.3.6.1.2.1.11.11) . . . 180
snmpInGenErrs (1.3.6.1.2.1.11.12) . . . 181
snmpInTotalReqVars (1.3.6.1.2.1.11.13) . . . 182
snmpInTotalSetVars (1.3.6.1.2.1.11.14) . . . 183
snmpInGetRequests (1.3.6.1.2.1.11.15) . . . 184
snmpInGetNexts (1.3.6.1.2.1.11.16) . . . 185
snmpInSetRequests (1.3.6.1.2.1.11.17) . . . 186
snmpInGetResponses (1.3.6.1.2.1.11.18) . . . 187
snmpInTraps (1.3.6.1.2.1.11.19) . . . 188
snmpOutTooBigs (1.3.6.1.2.1.11.20) . . . 189

snmpOutNoSuchNames (1.3.6.1.2.1.11.21) . . . 190
snmpOutBadValues (1.3.6.1.2.1.11.22) . . . 191
snmpOutGenErrs (1.3.6.1.2.1.11.24) . . . 192
snmpOutGetRequests (1.3.6.1.2.1.11.25) . . . 193
snmpOutGetNexts (1.3.6.1.2.1.11.26) . . . 194
snmpOutSetRequests (1.3.6.1.2.1.11.27) . . . 195
snmpOutGetResponses (1.3.6.1.2.1.11.28) . . . 196
snmpOutTraps (1.3.6.1.2.1.11.29) . . . 197
snmpEnableAuthenTraps (1.3.6.1.2.1.11.30) . . . 198
The ifXTable . . . 199
ifName (1.3.6.1.2.1.31.1.1.1.1) . . . 199
ifInMulticastPkts (1.3.6.1.2.1.31.1.1.1.2) . . . 200
ifInBroadcastPkts (1.3.6.1.2.1.31.1.1.1.3) . . . 201
ifOutMulticastPkts (1.3.6.1.2.1.31.1.1.1.4) . . . 202
ifOutBroadcastPkts (1.3.6.1.2.1.31.1.1.1.5) . . . 203
ifHighSpeed (1.3.6.1.2.1.31.1.1.1.15) . . . 204
ifPromiscuousMode (1.3.6.1.2.1.31.1.1.1.16) . . . 205
ifConnectorPresent (1.3.6.1.2.1.31.1.1.1.17) . . . 206
ifAlias (1.3.6.1.2.1.31.1.1.1.18) . . . 207
ifCounterDiscontinuityTime (1.3.6.1.2.1.31.1.1.1.19) . . . 208
ifTableLastChange (1.3.6.1.2.1.31.5) . . . 209

Chapter 5. Fibre Alliance MIB objects . . . 211

FA MIB definitions . . . 211
revisionNumber . . . 213
Connectivity unit group . . . 214
uNumber (1.3.6.1.3.94.1.1) . . . 214
systemURL (1.3.6.1.3.94.1.2) . . . 215
statusChangeTime (1.3.6.1.3.94.1.3) . . . 216
configurationChangeTime (1.3.6.1.3.94.1.4) . . . 217
connUnitTableChangeTime (1.3.6.1.3.94.1.5) . . . 218
Connectivity table . . . 219
connUnitId (1.3.6.1.3.94.1.6.1.1) . . . 219
connUnitGlobalId (1.3.6.1.3.94.1.6.1.2) . . . 220
connUnitType (1.3.6.1.3.94.1.6.1.3) . . . 221
connUnitNumports (1.3.6.1.3.94.1.6.1.4) . . . 222
connUnitState (1.3.6.1.3.94.1.6.1.5) . . . 223
connUnitStatus (1.3.6.1.3.94.1.6.1.6) . . . 224
connUnitProduct (1.3.6.1.3.94.1.6.1.7) . . . 225
connUnitSn (1.3.6.1.3.94.1.6.1.8) . . . 226
connUnitUpTime (1.3.6.1.3.94.1.6.1.9) . . . 227
connUnitUrl (1.3.6.1.3.94.1.6.1.10) . . . 228
connUnitDomainId (1.3.6.1.3.94.1.6.1.11 . . . 229
connUnitProxyMaster (1.3.6.1.3.94.1.6.1.12) . . . 230
connUnitPrincipal (1.3.6.1.3.94.1.6.1.13) . . . 231
connUnitNumSensors (1.3.6.1.3.94.1.6.1.14) . . . 232
connUnitStatusChangeTime (1.3.6.1.3.94.1.6.1.15) . . . 233
connUnitConfigurationChangeTime (1.3.6.1.3.94.1.6.1.16) . . . 234
connUnitNumRevs (1.3.6.1.3.94.1.6.1.17) . . . 235
connUnitNumZones (1.3.6.1.3.94.1.6.1.18) . . . 236
connUnitModuleId (1.3.6.1.3.94.1.6.1.19) . . . 237
connUnitName (1.3.6.1.3.94.1.6.1.20) . . . 238
connUnitInfo (1.3.6.1.3.94.1.6.1.21) . . . 239
connUnitControl (1.3.6.1.3.94.1.6.1.22) . . . 240
connUnitContact (1.3.6.1.3.94.1.6.1.23) . . . 241

connUnitLocation (1.3.6.1.3.94.1.6.1.24) . . . 242
connUnitEventFilter (1.3.6.1.3.94.1.6.1.25) . . . 243
connUnitNumEvents (1.3.6.1.3.94.1.6.1.26) . . . 244
connUnitMaxEvents (1.3.6.1.3.94.1.6.1.27) . . . 245
connUnitEventCurrID (1.3.6.1.3.94.1.6.1.28) . . . 246
connUnitFabricID (1.3.6.1.3.94.1.6.1.29) . . . 247
connUnitNumLinks (1.3.6.1.3.94.1.6.1.30) . . . 248
connUnitVendorId (1.3.6.1.3.94.1.6.1.31) . . . 249
Revision table . . . 250
connUnitRevsUnitId (1.3.6.1.3.94.1.7.1.1) . . . 250
connUnitRevsIndex (1.3.6.1.3.94.1.7.1.2) . . . 251
connUnitRevsRevId (1.3.6.1.3.94.1.7.1.3) . . . 252
connUnitRevsDescription (1.3.6.1.3.94.1.7.1.4) . . . 253
Sensor table . . . 253
connUnitSensorUnitId (1.3.6.1.3.94.1.8.1.1) . . . 254
connUnitSensorIndex (1.3.6.1.3.94.1.8.1.2) . . . 255
connUnitSensorName (1.3.6.1.3.94.1.8.1.3) . . . 256
connUnitSensorStatus (1.3.6.1.3.94.1.8.1.4) . . . 257
connUnitSensorInfo (1.3.6.1.3.94.1.8.1.5) . . . 258
connUnitSensorMessage (1.3.6.1.3.94.1.8.1.6) . . . 259
connUnitSensorType (1.3.6.1.3.94.1.8.1.7) . . . 260
connUnitSensorCharacteristic (1.3.6.1.3.94.1.8.1.8) . . . 261
Port table . . . 262
connUnitPortUnitId (1.3.6.1.3.94.1.10.1.1) . . . 262
connUnitPortIndex (1.3.6.1.3.94.1.10.1.2) . . . 263
connUnitPortType (1.3.6.1.3.94.1.10.1.3) . . . 264
connUnitPortFCClassCap (1.3.6.1.3.94.1.10.1.4) . . . 265
connUnitPortFCClassOp (1.3.6.1.3.94.1.10.1.5) . . . 266
connUnitPortState (1.3.6.1.3.94.1.10.1.6) . . . 267
connUnitPortStatus (1.3.6.1.3.94.1.10.1.7) . . . 268
connUnitPortTransmitterType (1.3.6.1.3.94.1.10.1.8) . . . 269
connUnitPortModuleType (1.3.6.1.3.94.1.10.1.9) . . . 270
connUnitPortWwn (1.3.6.1.3.94.1.10.1.10) . . . 271
connUnitPortFCId (1.3.6.1.3.94.1.10.1.11) . . . 272
connUnitPortSn (1.3.6.1.3.94.1.10.1.12) . . . 273
connUnitPortRevision (1.3.6.1.3.94.1.10.1.13) . . . 274
connUnitPortVendor (1.3.6.1.3.94.1.10.1.14) . . . 275
connUnitPortSpeed (1.3.6.1.3.94.1.10.1.15) . . . 276
connUnitPortControl (1.3.6.1.3.94.1.10.1.16) . . . 277
connUnitPortName (1.3.6.1.3.94.1.10.1.17) . . . 279
connUnitPortPhysicalNumber (1.3.6.1.3.94.1.10.1.18) . . . 280
connUnitPortStatObject (1.3.6.1.3.94.1.10.1.19) . . . 281
connUnitPortProtocolCap (1.3.6.1.3.94.1.10.1.20) . . . 282
connUnitPortProtocolOp (1.3.6.1.3.94.1.10.1.21) . . . 283
connUnitPortNodeWwn (1.3.6.1.3.94.1.10.1.22) . . . 284
connUnitPortHWState (1.3.6.1.3.94.1.10.1.23) . . . 285
Event table . . . 286
connUnitEventUnitId (1.3.6.1.3.94.1.11.1.1) . . . 286
connUnitEventIndex (1.3.6.1.3.94.1.11.1.2) . . . 287
connUnitEventId (1.3.6.1.3.94.1.11.1.3) . . . 288
connUnitREventTime (1.3.6.1.3.94.1.11.1.4) . . . 289
connUnitSEventTime (1.3.6.1.3.94.1.11.1.5) . . . 290
connUnitEventSeverity (1.3.6.1.3.94.1.11.1.6) . . . 291
connUnitEventType (1.3.6.1.3.94.1.11.1.7) . . . 292
connUnitEventObject (1.3.6.1.3.94.1.11.1.8) . . . 293

connUnitEventDescr (1.3.6.1.3.94.1.11.1.9) . . . 294
Link table . . . 295
connUnitLinkUnitId (1.3.6.1.3.94.1.12.1.1) . . . 295
connUnitLinkIndex (1.3.6.1.3.94.1.12.1.2) . . . 296
connUnitLinkNodeIdX (1.3.6.1.3.94.1.12.1.3) . . . 297
connUnitLinkPortNumberX (1.3.6.1.3.94.1.12.1.4) . . . 298
connUnitLinkPortWwnX (1.3.6.1.3.94.1.12.1.5) . . . 299
connUnitLinkNodeIdY (1.3.6.1.3.94.1.12.1.6) . . . 300
connUnitLinkPortNumberY (1.3.6.1.3.94.1.12.1.7) . . . 301
connUnitLinkPortWwnY (1.3.6.1.3.94.1.12.1.8) . . . 302
connUnitLinkAgentAddressY (1.3.6.1.3.94.1.12.1.9) . . . 303
connUnitLinkAgentAddressTypeY (1.3.6.1.3.94.1.12.1.10) . . . 304
connUnitLinkAgentPortY (1.3.6.1.3.94.1.12.1.11) . . . 305
connUnitLinkUnitTypeY (1.3.6.1.3.94.1.12.1.12) . . . 306
connUnitLinkConnIdY (1.3.6.1.3.94.1.12.1.13 . . . 307
connUnitLinkCurrIndex (1.3.6.1.3.94.1.12.1.14) . . . 308
Zone Table . . . 309
connUnitZoneIndex (1.3.6.1.3.94.1.13.1.1) . . . 309
connUnitZoneMemberIndex (1.3.6.1.3.94.1.13.1.2) . . . 310
connUnitZoneSetName (1.3.6.1.3.94.1.13.1.3) . . . 311
connUnitZoneSetNumZones (1.3.6.1.3.94.1.13.1.4) . . . 312
connUnitZoneName (1.3.6.1.3.94.1.13.1.5) . . . 313
connUnitZoneCapabilities (1.3.6.1.3.94.1.13.1.6) . . . 314
connUnitZoneEnforcementState (1.3.6.1.3.94.1.13.1.7) . . . 315
connUnitZoneAttributeBlock (1.3.6.1.3.94.1.13.1.8) . . . 316
connUnitZoneNumMembers (1.3.6.1.3.94.1.13.1.9) . . . 317
connUnitZoneMemberIdType (1.3.6.1.3.94.1.13.1.10) . . . 318
connUnitZoneMemberID (1.3.6.1.3.94.1.13.1.11) . . . 319
Zoning alias table . . . 320
connUnitZoningAliasIndex (1.3.6.1.3.94.1.14.1.1) . . . 320
connUnitZoningAliasMemberIndex (1.3.6.1.3.94.1.14.1.2) . . . 321
connUnitZoningAliasNumAliases (1.3.6.1.3.94.1.14.1.3) . . . 322
connUnitZoningAliasName (1.3.6.1.3.94.1.14.1.4) . . . 323
connUnitZoningAliasNumMembers (1.3.6.1.3.94.1.14.1.5) . . . 324
connUnitZoningAliasMemberIdType (1.3.6.1.3.94.1.14.1.6) . . . 325
connUnitZoningAliasMemberID (1.3.6.1.3.94.1.14.1.7) . . . 326
Port statistics table . . . 327
connUnitPortStatUnitId (1.3.6.1.3.94.4.5.1.1) . . . 327
connUnitPortStatIndex (1.3.6.1.3.94.4.5.1.2) . . . 328
connUnitPortStatCountError (1.3.6.1.3.94.4.5.1.3) . . . 329
connUnitPortStatCountTxObjects (1.3.6.1.3.94.4.5.1.4) . . . 330
connUnitPortStatCountRxObjects (1.3.6.1.3.94.4.5.1.5) . . . 331
connUnitPortStatCountTxElements (1.3.6.1.3.94.4.5.1.6) . . . 332
connUnitPortStatCountRxElements (1.3.6.1.3.94.4.5.1.7) . . . 333
connUnitPortStatCountBBCreditZero (1.3.6.1.3.94.4.5.1.8) . . . 334
connUnitPortStatCountInputBuffersFull (1.3.6.1.3.94.4.5.1.9) . . . 335
connUnitPortStatCountFBSYFrames (1.3.6.1.3.94.4.5.1.10) . . . 336
connUnitPortStatCountPBSYFrames (1.3.6.1.3.94.4.5.1.11) . . . 337
connUnitPortStatCountFRJTFrames (1.3.6.1.3.94.4.5.1.12) . . . 338
connUnitPortStatCountPRJTFrames (1.3.6.1.3.94.4.5.1.13) . . . 339
connUnitPortStatCountClass1RxFrames (1.3.6.1.3.94.4.5.1.14) . . . 340
connUnitPortStatCountClass1TxFrames (1.3.6.1.3.94.4.5.1.15) . . . 341
connUnitPortStatCountClass1FBSYFrames (1.3.6.1.3.94.4.5.1.16 . . . 342
connUnitPortStatCountClass1PBSYFrames (1.3.6.1.3.94.4.5.1.17) . . . 343
connUnitPortStatCountClass1FRJTFrames (1.3.6.1.3.94.4.5.1.18 . . . 344

connUnitPortStatCountClass1PRJTFrames (1.3.6.1.3.94.4.5.1.19) . . . 345
connUnitPortStatCountClass2RxFrames (1.3.6.1.3.94.4.5.1.20) . . . 346
connUnitPortStatCountClass2TxFrames (1.3.6.1.3.94.4.5.1.21) . . . 347
connUnitPortStatCountClass2FBSYFrames (1.3.6.1.3.94.4.5.1.22) . . . 348
connUnitPortStatCountClass2PBSYFrames (1.3.6.1.3.94.4.5.1.23) . . . 349
connUnitPortStatCountClass2FRJTFrames (1.3.6.1.3.94.4.5.1.24) . . . 350
connUnitPortStatCountClass2PRJTFrames (1.3.6.1.3.94.4.5.1.25) . . . 351
connUnitPortStatCountClass3RxFrames (1.3.6.1.3.94.4.5.1.26) . . . 352
connUnitPortStatCountClass3TxFrames (1.3.6.1.3.94.4.5.1.27) . . . 353
connUnitPortStatCountClass3Discards (1.3.6.1.3.94.4.5.1.28) . . . 354
connUnitPortStatCountRxMulticastObjects (1.3.6.1.3.94.4.5.1.29) . . . 355
connUnitPortStatCountTxMulticastObjects (1.3.6.1.3.94.4.5.1.30) . . . 356
connUnitPortStatCountRxBroadcastObjects (1.3.6.1.3.94.4.5.1.31) . . . 357
connUnitPortStatCountTxBroadcastObjects (1.3.6.1.3.94.4.5.1.32) . . . 358
connUnitPortStatCountRxLinkResets (1.3.6.1.3.94.4.5.1.33) . . . 359
connUnitPortStatCountTxLinkResets (1.3.6.1.3.94.4.5.1.34) . . . 360
connUnitPortStatCountNumberLinkResets (1.3.6.1.3.94.4.5.1.35) . . . 361
connUnitPortStatCountRxOfflineSequences (1.3.6.1.3.94.4.5.1.36) . . . 362
connUnitPortStatCountTxOfflineSequences (1.3.6.1.3.94.4.5.1.37) . . . 363
connUnitPortStatCountNumberOfflineSequences (1.3.6.1.3.94.4.5.1.38) . . . 364
connUnitPortStatCountLinkFailures (1.3.6.1.3.94.4.5.1.39) . . . 365
connUnitPortStatCountInvalidCRC (1.3.6.1.3.94.4.5.1.40) . . . 366
connUnitPortStatCountInvalidTxWords (1.3.6.1.3.94.4.5.1.41) . . . 367
connUnitPortStatCountPrimitiveSequenceProtocolErrors (1.3.6.1.3.94.4.5.1.42) . . . 368
connUnitPortStatCountLossofSignal (1.3.6.1.3.94.4.5.1.43) . . . 369
connUnitPortStatCountLossofSynchronization (1.3.6.1.3.94.4.5.1.44) . . . 370
connUnitPortStatCountInvalidOrderedSets (1.3.6.1.3.94.4.5.1.45) . . . 371
connUnitPortStatCountFramesTooLong (1.3.6.1.3.94.4.5.1.46) . . . 372
connUnitPortStatCountFramesTruncated (1.3.6.1.3.94.4.5.1.47) . . . 373
connUnitPortStatCountAddressErrors (1.3.6.1.3.94.4.5.1.48) . . . 374
connUnitPortStatCountDelimiterErrors (1.3.6.1.3.94.4.5.1.49) . . . 375
connUnitPortStatCountEncodingDisparityErrors (1.3.6.1.3.94.4.5.1.50) . . . 376
Simple Name Server Table . . . 377
connUnitSnsMaxEntry (1.3.6.1.3.94.5.1.1) . . . 377
connUnitSnsId (1.3.6.1.3.94.5.2.1.1.1) . . . 378
connUnitSnsPortIndex (1.3.6.1.3.94.5.2.1.1.2) . . . 379
connUnitSnsPortIdentifier (1.3.6.1.3.94.5.2.1.1.3) . . . 380
connUnitSnsPortName (1.3.6.1.3.94.5.2.1.1.4) . . . 381
connUnitSnsNodeName (1.3.6.1.3.94.5.2.1.1.5) . . . 382
connUnitSnsClassOfSvc (1.3.6.1.3.94.5.2.1.1.6) . . . 383
connUnitSnsNodeIPAddress (1.3.6.1.3.94.5.2.1.1.7) . . . 384
connUnitSnsProcAssoc (1.3.6.1.3.94.5.2.1.1.8) . . . 385
connUnitSnsFC4Type (1.3.6.1.3.94.5.2.1.1.9) . . . 386
connUnitSnsPortType (1.3.6.1.3.94.5.2.1.1.10) . . . 387
connUnitSnsPortIPAddress (1.3.6.1.3.94.5.2.1.1.11) . . . 388
connUnitSnsFabricPortName (1.3.6.1.3.94.5.2.1.1.12) . . . 389
connUnitSnsHardAddress (1.3.6.1.3.94.5.2.1.1.13) . . . 390
connUnitSnsSymbolicPortName (1.3.6.1.3.94.5.2.1.1.14) . . . 391
connUnitSnsSymbolicNodeName (1.3.6.1.3.94.5.2.1.1.15) . . . 392
Platform Table . . . 393
connUnitPlatformMaxEntry (1.3.6.1.3.94.5.1.2) . . . 393
connUnitPlatformIndex (1.3.6.1.3.94.5.2.2.1.1) . . . 394
connUnitPlatformNodeIndex (1.3.6.1.3.94.5.2.2.1.2) . . . 395
connUnitPlatformUnitID (1.3.6.1.3.94.5.2.2.1.3) . . . 396
connUnitPlatformName (1.3.6.1.3.94.5.2.2.1.4) . . . 397

connUnitPlatformType (1.3.6.1.3.94.5.2.2.1.6) . . . 398
connUnitPlatformLabel (1.3.6.1.3.94.5.2.2.1.7) . . . 399
connUnitPlatformDescription (1.3.6.1.3.94.5.2.2.1.8) . . . 400
connUnitPlatformLocation (1.3.6.1.3.94.5.2.2.1.9) . . . 401
connUnitPlatformManagementUrl (1.3.6.1.3.94.5.2.2.1.10) . . . 402
connUnitPlatformNumNodes (1.3.6.1.3.94.5.2.2.1.11) . . . 403
connUnitPlatformNodeName (1.3.6.1.3.94.5.2.2.1.12) . . . 404
Trap Table . . . 405
trapMaxClients (1.3.6.1.3.94.2.1) . . . 405
trapClientCount (1.3.6.1.3.94.2.2) . . . 406
trapRegIpAddress (1.3.6.1.3.94.2.3.1.1) . . . 407
trapRegPort (1.3.6.1.3.94.2.3.1.2) . . . 408
trapRegFilter (1.3.6.1.3.94.2.3.1.3) . . . 409
trapRegRowState (1.3.6.1.3.94.2.3.1.4) . . . 410
Related traps . . . 411
connUnitStatusChange (1.3.6.1.3.94.0.1) . . . 411
connUnitDeletedTrap (1.3.6.1.3.94.0.3) . . . 411
connUnitEventTrap (1.3.6.1.3.94.0.4) . . . 411
connUnitSensorStatusChange (1.3.6.1.3.94.0.5) . . . 413
connUnitPortStatusChange (1.3.6.1.3.94.0.6) . . . 413
coldStart . . . 413
authenticationFailure . . . 413

Chapter 6. Fabric Element MIB objects . . . 415

Fibre Channel FE MIB definitions . . . 415
Configuration group . . . 416
fcFeFabricName (1.3.6.1.2.1.75.1.1.1) . . . 416
fcFeElementName (1.3.6.1.2.1.75.1.1.2) . . . 417
fcFeModuleCapacity (1.3.6.1.2.1.75.1.1.3) . . . 418
Module table . . . 419
fcFeModuleDescr (1.3.6.1.2.1.75.1.1.4.1.2) . . . 419
fcFeModuleObjectID (1.3.6.1.2.1.75.1.1.4.1.3) . . . 420
fcFeModuleOperStatus (1.3.6.1.2.1.75.1.1.4.1.4) . . . 421
fcFeModuleLastChange (1.3.6.1.2.1.75.1.1.4.1.5) . . . 422
fcFeModuleFxPortCapacity (1.3.6.1.2.1.75.1.1.4.1.6) . . . 423
fcFeModuleName (1.3.6.1.2.1.75.1.1.4.1.7) . . . 424
FxPort configuration table . . . 425
fcFxPortName (1.3.6.1.2.1.75.1.1.5.1.2) . . . 425
fcFxPortFcphVersionHigh (1.3.6.1.2.1.75.1.1.5.1.3) . . . 426
fcFxPortFcphVersionLow (1.3.6.1.2.1.75.1.1.5.1.4) . . . 427
fcFxPortBbCredit (1.3.6.1.2.1.75.1.1.5.1.5) . . . 428
fcFxPortRxBufSize (1.3.6.1.2.1.75.1.1.5.1.6) . . . 429
fcFxPortRatov (1.3.6.1.2.1.75.1.1.5.1.7) . . . 430
fcFxPortEdtov (1.3.6.1.2.1.75.1.1.5.1.8) . . . 431
fcFxPortCosSupported (1.3.6.1.2.1.75.1.1.5.1.9) . . . 432
fcFxPortIntermixSupported (1.3.6.1.2.1.75.1.1.5.1.10) . . . 433
fcFxPortStackedConnMode (1.3.6.1.2.1.75.1.1.5.1.11) . . . 434
fcFxPortClass2SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.12) . . . 435
fcFxPortClass3SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.13) . . . 436
fcFxPortHoldTime (1.3.6.1.2.1.75.1.1.5.1.14) . . . 437
The Status group . . . 438
fcFxPortID (1.3.6.1.2.1.75.1.2.1.1.1) . . . 438
fcFxPortBbCreditAvailable (1.3.6.1.2.1.75.1.2.1.1.2) . . . 439
fcFxPortOperMode (1.3.6.1.2.1.75.1.2.1.1.3) . . . 440
fcFxPortAdminMode (1.3.6.1.2.1.75.1.2.1.1.4) . . . 441

FxPort physical level table 442
fcFxPortPhysAdminStatus (1.3.6.1.2.1.75.1.2.2.1.1) 442
fcFxPortPhysOperStatus (1.3.6.1.2.1.75.1.2.2.1.2) 443
fcFxPortPhysLastChange (1.3.6.1.2.1.75.1.2.2.1.3) 444
fcFxPortPhysRttov (1.3.6.1.2.1.75.1.2.2.1.4) 445
Fx Port fabric login table 446
fcFxPortFcphVersionAgreed (1.3.6.1.2.1.75.1.2.3.1.2) 446
fcFxPortNxPortBbCredit (1.3.6.1.2.1.75.1.2.3.1.3) 447
fcFxPortNxPortRxDataFieldSize (1.3.6.1.2.1.75.1.2.3.1.4) 448
fcFxPortCosSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.5) 449
fcFxPortIntermixSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.6) 450
fcFxPortStackedConnModeAgreed (1.3.6.1.2.1.75.1.2.3.1.7) 451
fcFxPortClass2SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.8) 452
fcFxPortClass3SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.9) 453
fcFxPortNxPortName (1.3.6.1.2.1.75.1.2.3.1.10) 454
fcFxPortConnectedNxPort (1.3.6.1.2.1.75.1.2.3.1.11) 455
fcFxPortBbCreditModel (1.3.6.1.2.1.75.1.2.3.1.12) 456
The Error group 457
fcFxPortLinkFailures (1.3.6.1.2.1.75.1.3.1.1.1) 457
fcFxPortSyncLosses (1.3.6.1.2.1.75.1.3.1.1.2) 458
fcFxPortSigLosses (1.3.6.1.2.1.75.1.3.1.1.3) 459
fcFxPortPrimSeqProtoErrors (1.3.6.1.2.1.75.1.3.1.1.4) 460
fcFxPortInvalidTxWords (1.3.6.1.2.1.75.1.3.1.1.5) 461
fcFxPortInvalidCrcs (1.3.6.1.2.1.75.1.3.1.1.6) 462
fcFxPortDelimiterErrors (1.3.6.1.2.1.75.1.3.1.1.7) 463
fcFxPortAddressIdErrors (1.3.6.1.2.1.75.1.3.1.1.8) 464
fcFxPortLinkResetIns (1.3.6.1.2.1.75.1.3.1.1.9) 465
fcFxPortLinkResetOuts (1.3.6.1.2.1.75.1.3.1.1.10) 466
fcFxPortOlsIns (1.3.6.1.2.1.75.1.3.1.1.11) 467
fcFxPortOlsOuts (1.3.6.1.2.1.75.1.3.1.1.12) 468
Class 1 accounting group 469
fcFxPortC1InFrames (1.3.6.1.2.1.75.1.4.1.1.1) 469
fcFxPortC1OutFrames (1.3.6.1.2.1.75.1.4.1.1.2) 470
fcFxPortC1InOctets (1.3.6.1.2.1.75.1.4.1.1.3) 471
fcFxPortC1OutOctets (1.3.6.1.2.1.75.1.4.1.1.4) 472
fcFxPortC1Discards (1.3.6.1.2.1.75.1.4.1.1.5) 473
fcFxPortC1FbsyFrames (1.3.6.1.2.1.75.1.4.1.1.6) 474
fcFxPortC1FrjtFrames (1.3.6.1.2.1.75.1.4.1.1.7) 475
fcFxPortC1InConnections (1.3.6.1.2.1.75.1.4.1.1.8) 476
fcFxPortC1OutConnections (1.3.6.1.2.1.75.1.4.1.1.9) 477
fcFxPortC1ConnTime (1.3.6.1.2.1.75.1.4.1.1.10) 478
Class 2 accounting group 479
fcFxPortC2InFrames (1.3.6.1.2.1.75.1.4.2.1.1) 479
fcFxPortC2OutFrames (1.3.6.1.2.1.75.1.4.2.1.2) 480
fcFxPortC2InOctets (1.3.6.1.2.1.75.1.4.2.1.3) 481
fcFxPortC2OutOctets (1.3.6.1.2.1.75.1.4.2.1.4) 482
fcFxPortC2Discards (1.3.6.1.2.1.75.1.4.2.1.5) 483
fcFxPortC2FbsyFrames (1.3.6.1.2.1.75.1.4.2.1.6) 484
fcFxPortC2FrjtFrames (1.3.6.1.2.1.75.1.4.2.1.7) 485
Class 3 accounting group 486
fcFxPortC3InFrames (1.3.6.1.2.1.75.1.4.3.1.1) 486
fcFxPortC3OutFrames (1.3.6.1.2.1.75.1.4.3.1.2) 487
fcFxPortC3InOctets (1.3.6.1.2.1.75.1.4.3.1.3) 488
fcFxPortC3OutOctets (1.3.6.1.2.1.75.1.4.3.1.4) 489
fcFxPortC3Discards (1.3.6.1.2.1.75.1.4.3.1.5) 490

Capability group 491
fcFxPortCapFcphVersionHigh (1.3.6.1.2.1.75.1.5.1.1.1) 491
fcFxPortCapFcphVersionLow (1.3.6.1.2.1.75.1.5.1.1.2) 492
fcFxPortCapBbCreditMax (1.3.6.1.2.1.75.1.5.1.1.3) 493
fcFxPortCapBbCreditMin (1.3.6.1.2.1.75.1.5.1.1.4) 494
fcFxPortCapRxDataFieldSizeMax (1.3.6.1.2.1.75.1.5.1.1.5) 495
fcFxPortCapRxDataFieldSizeMin (1.3.6.1.2.1.75.1.5.1.1.6) 496
fcFxPortCapCos (1.3.6.1.2.1.75.1.5.1.1.7) 497
fcFxPortCapIntermix (1.3.6.1.2.1.75.1.5.1.1.8) 498
fcFxPortCapStackedConnMode (1.3.6.1.2.1.75.1.5.1.1.9) 499
fcFxPortCapClass2SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.10) 500
fcFxPortCapClass3SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.11) 501
fcFxPortCapHoldTimeMaxv (1.3.6.1.2.1.75.1.5.1.1.12) 502
fcFxPortCapHoldTimeMin (1.3.6.1.2.1.75.1.5.1.1.13) 503

Chapter 7. Private enterprise MIB objects 505

Private Enterprise MIB definitions 505
fcQxPortPhysAdminStatus (1.3.6.1.4.1.1663.1.3.10.1.1.3) 505
fcQxPortPhysOperStatus (1.3.6.1.4.1.1663.1.3.10.1.1.4) 507
Related Traps 508
qlSB2PortLinkDown (qLogicExperimental 0 10) 508
qlSB2PortLinkUp (qLogicExperimental 0 11) 508
qlconnUnitAddedTrap (qLogicExperimental 0 12) 508

Chapter 8. Firmware download MIB objects 509

Firmware download MIB definitions 509
qlgcChFwOpResult (1.3.6.1.4.1.3873.3.1.1.2.1) 509
qlgcChFwOpRequest (1.3.6.1.4.1.3873.3.1.1.2.2) 510
qlgcChFwDwldHostAddrType (1.3.6.1.4.1.3873.3.1.1.2.3) 511
qlgcChFwDwldHostAddr (1.3.6.1.4.1.3873.3.1.1.2.4) 512
qlgcChFwDwldHostPort (1.3.6.1.4.1.3873.3.1.1.2.5) 513
qlgcChFwDwldPathName (1.3.6.1.4.1.3873.3.1.1.2.6) 514
qlgcChFwDwldFileName (1.3.6.1.4.1.3873.3.1.1.2.7) 515
qlgcChFwResetMethod (1.3.6.1.4.1.3873.3.1.1.2.8) 516
Before you call 517
Using the documentation 517
Getting help and information from the World Wide Web 518
Software service and support 518
Hardware service and support 518
IBM Taiwan product service 518
Trademarks 520
Important notes 520
Documentation format 521

Glossary 523

Index 527

Chapter 1. Simple Network Management Protocol

This guide describes the support for Simple Network Management Protocol (SNMP) and how to use SNMP to manage and monitor the IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru.

Related documentation

The product documentation for your specific IBM Flex System network switch, pass-thru module, or chassis might contain additional, more-detailed troubleshooting information. For the most up-to-date product documentation for all of your IBM Flex System products, go to the IBM Flex System Information Center at http://publib.boulder.ibm.com/infocenter/flexsys/information/index.jsp.

The following documentation contains important, useful information to help you with the setup, installation, configuration, operation, and troubleshooting processes for these devices. This documentation is preloaded on the IBM Flex System Manager and is also available at http://publib.boulder.ibm.com/infocenter/flexsys/information/index.jsp:

- IBM Flex System network device *User's Guides*

  Each type of network adapter has a customized *Installation and User's Guide* that contains detailed information about the expansion card, which is compatible with the 8 Gb switches. These switches contain connectors for the compute nodes in which the network adapter is installed.

- *IBM Flex System Enterprise Chassis Installation and Service Guide*

  Each type of IBM Flex System chassis has a customized *Installation and Service Guide*.

- IBM Flex System compute node *Installation and Service Guides*

  Each type of compute node has a customized *Installation and Service Guide*.

- IBM *Notices for Network Devices* CD

  This CD ships with networking products (adapters, switches, and pass-thru modules).

- *IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru Installation and User's Guide*

  This document contains instructions for setting up, installing, removing, configuring, and troubleshooting the switch.

- *IBM Flex System FC3171 8 Gb SAN Switch Command Line Interface User's Guide*

  This document explains how to manage the SAN switch using the CLI.

- *IBM Flex System FC3171 8 Gb Pass-thru Command Line Interface User's Guide*

  This document explains how to manage the pass-thru module using the CLI.

- *IBM Flex System FC3171 8 Gb SAN Switch QuickTools User's Guide*

  This document explains how to manage the SAN switch using the QuickTools application.

- *IBM Flex System FC3171 8 Gb Pass-thru QuickTools User's Guide*

  This document explains how to manage the pass-thru module using the QuickTools application.

- *IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru CIM Agent Reference Guide*

  This document describes how the Common Interface Model (CIM) Agent functions as an implementation of the Storage Management Initiative (SMI)-Specification 1.1

- *IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru Event Message Reference Guide*

  This document lists the event messages for the IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru.

The updated IBM Flex System documentation is available on the IBM Flex System Manager and from the IBM Flex System Information Center at http://publib.boulder.ibm.com/infocenter/flexsys/information/index.jsp.

Notices and statements in this document

The caution and danger statements in this document are also in the multilingual *Safety Information* document, which is provided on the IBM *Notices for Network Device* CD. Each statement is numbered for reference to the corresponding statement in your language in the *Safety Information* document.

The following notices and statements are used in this document:

- **Note:** These notices provide important tips, guidance, or advice.
- **Important:** These notices provide information or advice that might help you avoid inconvenient or problem situations.
- **Attention:** These notices indicate potential damage to programs, devices, or data. An attention notice is placed just before the instruction or situation in which damage could occur.
- **Caution:** These statements indicate situations that can be potentially hazardous to you. A caution statement is placed just before the description of a potentially hazardous procedure step or situation.
- **Danger:** These statements indicate situations that can be potentially lethal or extremely hazardous to you. A danger statement is placed just before the description of a potentially lethal or extremely hazardous procedure step or situation.

Chapter 2. SNMP overview

Simple Network Management Protocol is the protocol governing network management and monitoring of network devices. This Simple Network Management Protocol Reference Guide describes how to use SNMP to manage and monitor the IBM FC3171 switch. Specifically, this guide describes the SNMP agent that resides on the switch.

The following topics are covered in this section:

- SNMP interface objectives
- Manager and agent
- Traps
- Management information bases (MIBs)
- User datagram protocol (UDP)
- Numbering system conventions

SNMP interface objectives

The objectives of the SNMP Interface are as follows:

- Connect to the SNMP agent that resides on the switch using a management workstation.
- Support of Fabric Element Management Information Bases (FE-MIB) (rfc2837) and Fibre Alliance Management Information Bases (FA-MIB) draft.
- Support of version 1 and 2 traps.
- The SNMP agent supports SNMPv1and SNMPv2c.

Manager and agent

The two primary elements of SNMP are:

- **Manager**—the application that runs on the management workstation.
- **Agent**—the daemon application that runs on the switch.

The Manager is the application through which the network administrator performs network management functions. The SNMP agent is the direct interface on the switch for any SNMP manager connecting to the switch using the SNMP protocol, as shown in Figure 1. The agent will be started by the script file(s) responsible for switch initialization when the switch powers up or when the switch is reset.

When an SNMP request arrives at the agent, the agent will compose a message and pass it on to Switch Management to process the message and provide a response to the agent. The agent then provides a response to the originator of the SNMP request. The SNMP agent does not have direct access to the internal database of the switch.

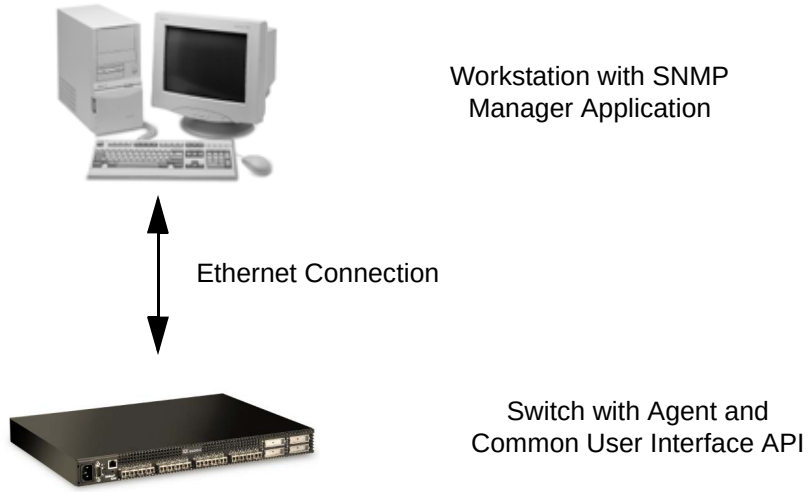

Figure 1. SNMP interface architecture

Traps

Traps are notification messages sent from the switch to a registered manager when a change of state occurs within the switch. A change of state can be an alarm condition or simply a configuration change.

The Fibre Alliance MIB defines a trap table configurable through SNMP. A trap table may have up to 5 entries, and can be configured using the SNMP Manager or Enterprise Fabric Suite graphical user interface. The same trap table information is available to both SNMP Manager and Enterprise Fabric Suite.

A trap event is reported when the incoming error has a severity level less than or equal to the configured severity level. The trap event types and trap severity levels are listed in Table 1. Refer to Table 3 for information on specific traps.

Table 1. Trap Severity Levels

Event type	**Severity level**
Unknown	1
Emergency	2
Alert	3
Critical	4
Error	5
Warning	6
Notify	7
Info	8
Debug	9
Mark	10

Management information base

Management information bases (MIBs) define the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each definition written in the MIB. It is not the actual database itself; it is implementation dependant. Definition of the MIB conforms to the Structure of Management Information (SMI) given in Request For Comment (RFC) 1155. The latest Internet MIB is given in RFC 1213, and is sometimes called MIB-II.

User datagram protocol

The IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru supports the following User Datagram Protocol (UDP) settings:

- Agents “listen” on UDP port 161.
- Responses are sent back to the originating Network Management Station (NMS) port from a dynamic port, although many agents use port 161 also for this target.
- The maximum SNMP message size is 65507 octets (maximum UDP message size).
- The minimum receive packet size for SNMP implementations is 484 octets in length.
- Agent and Network Monitoring Systems are responsible for determining error recovery.

Numbering system conventions

The conventions for numbering systems in this guide are as follows:

- Decimal = 101
- Hexadecimal = 0x101
- Binary = 101b

Chapter 3. Configuring a switch

This chapter describes how to configure an IBM FC3171 switch to support SNMP. The following topics are covered:

- System specifications and requirements
- Configuring a switch using the Telnet command line interface
- Configuring a switch using the QuickTools web applet

System Specifications and requirements

- The IBM FC3171 switch supports SNMPv1 and SNMPv2c.
- Version 1 and 2 traps are supported.
- Hardware—one out-of-band Ethernet connection is required.
- Software—one switch management software application allows you to:
  - Monitor and control the switch.
  - Read, write, and receive trap information, if supported.
- Ports on the switch reserved for SNMP:
  - Port 161 is not configurable, and is used for the standard SNMP commands.
  - Port 162 is configurable and is the default port used for traps.
- One or more in-band IBM FC3171 switches can be managed by an out-of-band IBM FC3171 switch acting as a proxy switch.
- An IBM FC3171 switch can only act as a proxy for another IBM FC3171 switch.
- The IBM FC3171 switch proxy capability can be disabled.

Configuring a switch using the command line interface

The Telnet command line interface offers a convenient way to change SNMP parameters. SNMP parameter defaults are preset during manufacturing. For security purposes, these default values should be changed. For specific information about SNMP parameters, refer to the SNMP Configuration section in the corresponding *IBM Flex System FC3171 8 Gb SAN Switch Command Line Interface User's Guide*. To configure a switch using the command line interface, do the following.

Press **ENTER** to accept the default value for each parameter.

```
IBM8Gb (admin) #> set setup snmp
A list of attributes with formatting and current values will follow.
Enter a new value or simply press the ENTER key to accept the current
value.
If you wish to terminate this process before reaching the end of the
attributes for the category being processed, press 'q' or 'Q' and the
ENTER key to do so.
  If you wish to terminate the configuration process completely, press
'qq' or
  'QQ' and the ENTER key to do so.

  SNMP System Configuration - may optionally use 'set setup snmp common'
  command.

  Current Values:
    SnmpEnabled      True
    Contact          <sysContact undefined>
    Location         <sysLocation undefined>
    ReadCommunity    public
    WriteCommunity   private
    AuthFailureTrap  False
    ProxyEnabled     True
    SNMPv3Enabled    False

  New Value (press ENTER to not specify value, 'q' to quit):
    SnmpEnabled      (True / False)         :
    Contact          (string, max=64 chars) :
    Location         (string, max=64 chars) :
    ReadCommunity    (string, max=32 chars) :
    WriteCommunity   (string, max=32 chars) :
    AuthFailureTrap  (True / False)         :
    ProxyEnabled     (True / False)         :
    SNMPv3Enabled    (True / False)         :

  SNMP Trap 1 Configuration - may optionally use 'set setup snmp trap 1'
  command.

  Current Values:
    Trap1Enabled    False
    Trap1Address    10.0.0.254
    Trap1Port       5001
    Trap1Severity   info
    Trap1Version    2
    Trap1Community  public

  New Value (press ENTER to not specify value, 'q' to quit):
    Trap1Enabled    (True / False)                    :
    Trap1Address    (hostname, IPv4, or IPv6 Address) :
    Trap1Port       (decimal value, 1-65535)          :
    Trap1Severity   (select a severity level)
                    1=unknown     6=warning
                    2=emergency   7=notify
```

```
                3=alert       8=info
                4=critical    9=debug
                5=error       10=mark           :
  Trap1Version    (1 / 2)                         :
  Trap1Community  (string, max=32 chars)          :

SNMP Trap 2 Configuration - may optionally use 'set setup snmp trap 2'
command.

Current Values:
  Trap2Enabled    False
  Trap2Address    10.20.43.231
  Trap2Port       162
  Trap2Severity   info
  Trap2Version    2
  Trap2Community  public

New Value (press ENTER to not specify value, 'q' to quit):
  Trap2Enabled    (True / False)                  :
  Trap2Address    (hostname, IPv4, or IPv6 Address) :
  Trap2Port       (decimal value, 1-65535)        :
  Trap2Severity   (select a severity level)
                  1=unknown     6=warning
                  2=emergency   7=notify
                  3=alert       8=info
                  4=critical    9=debug
                  5=error       10=mark           :
  Trap2Version    (1 / 2)                         :
  Trap2Community  (string, max=32 chars)          :

SNMP Trap 3 Configuration - may optionally use 'set setup snmp trap 3'
command.

Current Values:
  Trap3Enabled    False
  Trap3Address    10.20.33.231
  Trap3Port       162
  Trap3Severity   warning
  Trap3Version    2
  Trap3Community  public

New Value (press ENTER to not specify value, 'q' to quit):
  Trap3Enabled    (True / False)                  :
  Trap3Address    (hostname, IPv4, or IPv6 Address) :
  Trap3Port       (decimal value, 1-65535)        :
  Trap3Severity   (select a severity level)
                  1=unknown     6=warning
                  2=emergency   7=notify
                  3=alert       8=info
                  4=critical    9=debug
                  5=error       10=mark           :
  Trap3Version    (1 / 2)                         :
```

```
  Trap3Community  (string, max=32 chars)           :

SNMP Trap 4 Configuration - may optionally use 'set setup snmp trap 4'
command.

Current Values:
  Trap4Enabled    False
  Trap4Address    0.0.0.0
  Trap4Port       162
  Trap4Severity   warning
  Trap4Version    2
  Trap4Community  public

New Value (press ENTER to not specify value, 'q' to quit):
  Trap4Enabled    (True / False)                   :
  Trap4Address    (hostname, IPv4, or IPv6 Address) :
  Trap4Port       (decimal value, 1-65535)         :
  Trap4Severity   (select a severity level)
                  1=unknown     6=warning
                  2=emergency   7=notify
                  3=alert       8=info
                  4=critical    9=debug
                  5=error       10=mark           :
  Trap4Version    (1 / 2)                          :
  Trap4Community  (string, max=32 chars)           :

SNMP Trap 5 Configuration - may optionally use 'set setup snmp trap 5'
command.

Current Values:
  Trap5Enabled    False
  Trap5Address    0.0.0.0
  Trap5Port       162
  Trap5Severity   warning
  Trap5Version    2
  Trap5Community  public

New Value (press ENTER to not specify value, 'q' to quit):
  Trap5Enabled    (True / False)                   :
  Trap5Address    (hostname, IPv4, or IPv6 Address) :
  Trap5Port       (decimal value, 1-65535)         :
  Trap5Severity   (select a severity level)
                  1=unknown     6=warning
                  2=emergency   7=notify
                  3=alert       8=info
                  4=critical    9=debug
                  5=error       10=mark           :
  Trap5Version    (1 / 2)                          :
  Trap5Community  (string, max=32 chars)           :

  Do you want to save and activate this snmp setup? (y/n): [n]
SNMP setup NEITHER saved NOR activated.
```

Configuring a switch Using QuickTools

To configure an IBM FC3171 switch using QuickTools, use the SNMP Properties, Switch Properties, and Network Properties dialogs. For specific information, refer to the *QuickTools Switch Management User Guide*.

Chapter 4. MIB-II objects

This chapter covers the implementation details for the MIB-II on the IBM FC3171 switch. A MIB defines the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each definition written in the MIB. It is not the actual database itself; it is implementation dependant. Definition of the MIB conforms to the SMI given in RFC 1155. The latest Internet MIB is given in RFC 1213, and is sometimes called MIB-II.

Groups in MIB-II

Refer the Table 2 for the syntax for MIB-II Groups.

Table 2. MIB-II Groups

Group	Syntax
system	OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces	OBJECT IDENTIFIER ::= { mib-2 2 }
at	OBJECT IDENTIFIER ::= { mib-2 3 }
ip	OBJECT IDENTIFIER ::= { mib-2 4 }
icmp	OBJECT IDENTIFIER ::= { mib-2 5 }
tcp	OBJECT IDENTIFIER ::= { mib-2 6 }
udp	OBJECT IDENTIFIER ::= { mib-2 7 }
snmp	OBJECT IDENTIFIER ::= { mib-2 11 }

System group

Implementation of the System group is mandatory for all systems. If an agent is not configured to have a value for any of these variables, a string of length 0 is returned.

sysDescr (1.3.6.1.2.1.1.1)

A textual description of the entity. This value should include the full name and version identification of the system's hardware type, operating-system, and networking software. It is mandatory that this only contain printable American Standard Code for Information Interchange (ASCII) characters.

Syntax DisplayString (SIZE (0..255))

Access read-only

Status Mandatory

Return Value `IBM Flex System FC3171 8 Gb SAN Switch` or
`IBM Flex System FC3171 8 Gb SAN Switch Pass-thru`

sysObjectID (1.3.6.1.2.1.1.2)

The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprise subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining `what kind of box' is being managed. For example, if vendor `Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its `Fred Router'.

Syntax OBJECT IDENTIFIER

Access read-only

Status mandatory

Return Value `1.3.6.1.4.1.3873.1.33`

sysUpTime (1.3.6.1.2.1.1.3)

The time, in hundredths of a second, since the network management portion of the system was last re-initialized.

Syntax TimeTicks

Access read-only

Status mandatory

Return Value The time since the switch was powered on, or last reset (reset, hardreset, or hotreset) was executed. For example, 3 days 21 hours, 5 minutes, and 26.84 seconds. The value will roll over after approximately 497 days of continuous up time.

sysContact (1.3.6.1.2.1.1.4)

The textual identification of the contact person for this managed Node, together with information on how to contact this person.

Syntax DisplayString (SIZE (0..255))

Access read-write

Status mandatory

Return Value Default: `<sysContact undefined>`. The string size is limited to a maximum of 64.

sysName (1.3.6.1.2.1.1.5)

An administratively assigned name for this managed Node. By convention, this is the Node's fully qualified domain name.

Syntax DisplayString (SIZE (0..255))

Access read-write

Status mandatory

Return Value Default: `IBM8Gb`

sysLocation (1.3.6.1.2.1.1.6)

The physical location of this Node, such as telephone closet and 3rd floor.

Syntax DisplayString (SIZE (0..255))

Access read-write

Status mandatory

Return Value Default: `<sysLocation undefined>`. The string size is limited to a maximum of 64.

sysServices (1.3.6.1.2.1.1.7)

A value that indicates the set of services that this entity primarily offers.The value is a sum. This sum initially takes the value zero. Then, for each layer L in the range 1 through 7 that this Node performs transactions for, 2 raised to (L - 1) is added to the sum. For example, a Node that performs primarily routing functions would have a value of 4 (2^(3-1)). In contrast, a Node that is a host offering application services would have a value of 72 (2^(4-1) + 2^(7-1)).

Syntax INTEGER (0..127)

Access read-only

Status mandatory

Return Value Default: 2

The Interfaces Group

Implementation of the Interfaces group is mandatory for all systems.

ifNumber (1.3.6.1.2.1.2.1)

The number of network interfaces (regardless of their current state) present on this system.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Default: 2

The interfaces table

The Interfaces table contains information on the entity's interfaces. Each interface is thought of as being attached to a `subnetwork'. This term should not be confused with `subnet' that refers to an addressing partitioning scheme used in the Internet suite of protocols.

ifIndex (1.3.6.1.2.1.2.2.1.1)

A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization. The range will be non-contiguous.

Syntax INTEGER

Access read-only

Status mandatory

ifDescr (1.3.6.1.2.1.2.2.1.2)

A textual string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.

Syntax DisplayString (SIZE (0..255))

Access read-only

Status mandatory

ifType (1.3.6.1.2.1.2.2.1.3)

The type of interface distinguished according to the physical/link protocol(s) immediately `below' the network layer in the protocol stack.

Syntax INTEGER

Access read-only

Status mandatory

ifMtu (1.3.6.1.2.1.2.2.1.4)

The size of the largest datagram that can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

Syntax INTEGER

Access read-only

Status mandatory

ifSpeed (1.3.6.1.2.1.2.2.1.5)

An estimate of the interface's current bandwidth in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object, then this object reports its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed.

Syntax Gauge

Access read-only

Status mandatory

ifPhysAddress (1.3.6.1.2.1.2.2.1.6)

The interface's address at the protocol layer immediately “below” the network layer in the protocol stack. For interfaces that do not have such an address, such as a serial line, this object should contain an octet string of zero length.

Syntax PhysAddress

Access read-only

Status mandatory

ifAdminStatus (1.3.6.1.2.1.2.2.1.7)

The desired state of the interface. The testing(3) state indicates that no operational packets can be passed.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Writes not supported.

ifOperStatus (1.3.6.1.2.1.2.2.1.8)

The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed.

Syntax INTEGER

Access read-only

Status mandatory

ifLastChange (1.3.6.1.2.1.2.2.1.9)

The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.

Syntax TimeTicks

Access read-only

Status mandatory

ifInOctets (1.3.6.1.2.1.2.2.1.10)

The total number of octets received on the interface, including framing characters.

Syntax Counter

Access read-only

Status mandatory

ifInUcastPkts (1.3.6.1.2.1.2.2.1.11)

The number of subnetwork-unicast packets delivered to a higher-layer protocol.

Syntax Counter

Access read-only

Status mandatory

ifInNUcastPkts (1.3.6.1.2.1.2.2.1.12)

The number of non-unicast (that is, subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.

Syntax Counter

Access read-only

Status mandatory

ifInDiscards (1.3.6.1.2.1.2.2.1.13)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

Syntax Counter

Access read-only

Status mandatory

ifInErrors (1.3.6.1.2.1.2.2.1.14)

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Syntax Counter

Access read-only

Status mandatory

ifInUnknownProtos (1.3.6.1.2.1.2.2.1.15)

The number of packets received from the interface that were discarded because of an unknown or unsupported protocol.

Syntax Counter

Access read-only

Status mandatory

ifOutOctets (1.3.6.1.2.1.2.2.1.16)

The total number of octets transmitted out of the interface, including framing characters.

Syntax Counter

Access read-only

Status mandatory

ifOutUcastPkts (1.3.6.1.2.1.2.2.1.17)

The total number of packets that higher level protocols requested be transmitted to a subnetwork unicast address, including those that were discarded or not sent.

Syntax Counter

Access read-only

Status mandatory

ifOutNUcastPkts (1.3.6.1.2.1.2.2.1.18)

The total number of packets that higher level protocols requested be transmitted to a non-unicast (subnetwork broadcast or subnetwork multicast) address, including those that were discarded or not sent.

Syntax Counter

Access read-only

Status mandatory

ifOutDiscards (1.3.6.1.2.1.2.2.1.19)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Syntax Counter

Access read-only

Status mandatory

ifOutErrors (1.3.6.1.2.1.2.2.1.20)

The number of outbound packets that could not be transmitted because of errors.

Syntax Counter

Access read-only

Status mandatory

ifOutQLen (1.3.6.1.2.1.2.2.1.21)

The length (in packets) of the output packet queue.

Syntax Gauge

Access read-only

Status mandatory

ifSpecific (1.3.6.1.2.1.2.2.1.22)

A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an Ethernet, then the value of this object refers to a document that defines objects specific to Ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 (Abstract Syntax Notation) and BER must be able to generate and recognize this value.

Syntax OBJECT IDENTIFIER

Access read-only

Status mandatory

The Address Translation Group

Implementation of the Address Translation group is mandatory for all systems. However, this group is deprecated by MIB-II. That is, it is being included solely for compatibility with MIB-I Nodes, and will most likely be excluded from MIB-III Nodes. From MIB-II and onwards, each network protocol group contains its own address translation tables.

The Address Translation group contains one table that is the union across all interfaces of the translation tables for converting a NetworkAddress (for example, an IP address) into a subnetwork-specific address. For lack of a better term, this document refers to such a subnetwork-specific address as a `physical' address.

Examples of such translation tables are for: broadcast media where ARP is in use, the translation table is equivalent to the ARP cache, or on an X.25 network where non-algorithmic translation to X.121 addresses is required. The translation table contains the NetworkAddress to X.121 address equivalences.

atIfIndex (1.3.6.1.2.1.3.1.1.1)

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Syntax INTEGER

Access read-write

Status deprecated

atPhysAddress (1.3.6.1.2.1.3.1.1.2)

The media-dependent "physical" address. Setting this object to a null string (one of zero length) has the effect of invalidating the corresponding entry in the atTable object. That is, it effectively disassociates the interface identified with the entry from the mapping identified with the entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management workstations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant atPhysAddress object.

Syntax PhysAddress

Access read-write

Status deprecated

atNetAddress (1.3.6.1.2.1.3.1.1.3)

The NetworkAddress corresponding to the media-dependent `physical' address.

Syntax NetworkAddress

Access read-write

Status deprecated

The IP Group

Implementation of the IP group is mandatory for all systems.

ipForwarding (1.3.6.1.2.1.4.1)

The indication of whether this entity is acting as an IP Gateway with respect to the forwarding of datagrams received by, but not addressed to, this entity. IP Gateways forward datagrams; IP hosts do not (except those source-routed from the host).

For some managed Nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to change this object to an inappropriate value.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Returns forwarding (1). Writes not supported.

ipDefaultTTL (1.3.6.1.2.1.4.2)

The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity whenever a TTL value is not supplied by the transport layer protocol.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Returns 64 (0x40). Writes not supported.

ipInReceives (1.3.6.1.2.1.4.3)

The total number of input datagrams received from interfaces, including those received in error.

Syntax Counter

Access read-only

Status mandatory

ipInHdrErrors (1.3.6.1.2.1.4.4)

The number of input datagrams discarded due to errors in their IP headers. These include bad checksums, version number mismatch, other format errors, time-to-live exceeded, and errors discovered in processing their IP options.

Syntax Counter

Access read-only

Status mandatory

ipInAddrErrors (1.3.6.1.2.1.4.5)

The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported Classes (for example, Class E). For entities that are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

Syntax Counter

Access read-only

Status mandatory

ipForwDatagrams (1.3.6.1.2.1.4.6)

The number of input datagrams for which this entity was not their final IP destination. As a result, an attempt was made to find a route to forward them to that final destination. In entities that do not act as IP Gateways, this counter will include only those packets that were Source Routed from this entity, and the Source Route option processing was successful.

Syntax Counter

Access read-only

Status mandatory

ipInUnknownProtos (1.3.6.1.2.1.4.7)

The number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

Syntax Counter

Access read-only

Status mandatory

ipInDiscards (1.3.6.1.2.1.4.8)

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but were discarded (for example, for lack of buffer space). This counter does not include any datagrams discarded while awaiting reassembly.

Syntax Counter

Access read-only

Status mandatory

ipInDelivers (1.3.6.1.2.1.4.9)

The total number of input datagrams successfully delivered to IP user protocols (including ICMP).

Syntax Counter

Access read-only

Status mandatory

ipOutRequests (1.3.6.1.2.1.4.10)

The total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission. This counter does not include any datagrams counted in ipForwDatagrams.

Syntax Counter

Access read-only

Status mandatory

ipOutDiscards (1.3.6.1.2.1.4.11)

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but were discarded (for example, for lack of buffer space). This counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.

Syntax Counter

Access read-only

Status mandatory

ipOutNoRoutes (1.3.6.1.2.1.4.12)

The number of IP datagrams discarded because no route could be found to transmit them to their destination. This counter includes any packets counted in ipForwDatagrams that meet this “no-route” criterion. This includes any datagrams that a host cannot route because all of its default gateways are down.

Syntax Counter

Access read-only

Status mandatory

ipReasmTimeout (1.3.6.1.2.1.4.13)

The maximum number of seconds for which received fragments are held while they are awaiting reassembly at this entity.

Syntax INTEGER

Access read-only

Status mandatory

ipReasmReqds (1.3.6.1.2.1.4.14)

The number of IP fragments received that needed to be reassembled at this entity.

Syntax Counter

Access read-only

Status mandatory

ipReasmOKs (1.3.6.1.2.1.4.15)

The number of IP datagrams successfully reassembled.

Syntax Counter

Access read-only

Status mandatory

ipReasmFails (1.3.6.1.2.1.4.16)

The number of failures detected by the IP reassembly algorithm for example, timed out, errors). This is not necessarily a count of discarded IP fragments, since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

Syntax Counter

Access read-only

Status mandatory

ipFragOKs (1.3.6.1.2.1.4.17)

The number of IP datagrams that have been successfully fragmented at this entity.

Syntax Counter

Access read-only

Status mandatory

ipFragFails (1.3.6.1.2.1.4.18)

The number of IP datagrams that have been discarded because they needed to be fragmented at this entity, but could not because their Don't Fragment flag was set.

Syntax Counter

Access read-only

Status mandatory

ipFragCreates (1.3.6.1.2.1.4.19)

The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

Syntax Counter

Access read-only

Status mandatory

The IP address table

The IP address table contains this entity's IP addressing information.

ipAdEntAddr (1.3.6.1.2.1.4.20.1.1)

The IP address to which this entry's addressing information pertains.

Syntax IpAddress

Access read-only

Status mandatory

ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2)

The index value that uniquely identifies the interface to which this entry applies. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Syntax INTEGER

Access read-only

Status mandatory

ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3)

The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.

Syntax IpAddress

Access read-only

Status mandatory

ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4)

The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcasts addresses used by the entity on this (logical) interface.

Syntax INTEGER

Access read-only

Status mandatory

ipAdEntReasmMaxSize (1.3.6.1.2.1.4.20.1.5)

The size of the largest IP datagram that this entity can reassemble from incoming IP fragmented datagrams received on this interface.

Syntax INTEGER (0..65535)

Access read-only

Status mandatory

The IP Routing Table

The IP routing table contains an entry for each route presently known to this entity.

ipRouteDest (1.3.6.1.2.1.4.21.1.1)

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

Syntax IpAddress

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteIfIndex (1.3.6.1.2.1.4.21.1.2)

The index value that uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3)

The primary routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMetric2 (1.3.6.1.2.1.4.21.1.4)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMetric3 (1.3.6.1.2.1.4.21.1.5)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMetric4 (1.3.6.1.2.1.4.21.1.6)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteNextHop (1.3.6.1.2.1.4.21.1.7)

The IP address of the next hop of this route. In the case of a route bound to an interface that is realized from a broadcast media, the value of this field is the agent's IP address on that interface.

Syntax IpAddress

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteType (1.3.6.1.2.1.4.21.1.8)

The type of route. The values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with the entry from the route identified with the entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteProto (1.3.6.1.2.1.4.21.1.9)

The routing mechanism through which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.

Syntax INTEGER

Access read-only

Status mandatory

ipRouteAge (1.3.6.1.2.1.4.21.1.10)

The number of seconds since this route was last updated or otherwise determined to be correct. No semantics of `too old' can be implied except through knowledge of the routing protocol by which the route was learned.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMask (1.3.6.1.2.1.4.21.1.11)

Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field.

Syntax IpAddress

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteMetric5 (1.3.6.1.2.1.4.21.1.12)

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipRouteInfo (1.3.6.1.2.1.4.21.1.13)

A reference to MIB definitions specific to the particular routing protocol that is responsible for this route, as determined by the value specified in the route's ipRouteProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier. Any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.

Syntax OBJECT IDENTIFIER

Access read-only

Status mandatory

The IP Address Translation Table

The IP address translation table contain the IpAddress to`physical' address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 has an algorithmic method). If all interfaces are of this type, then the Address Translation table is empty, that is, has zero entries.

ipNetToMediaIfIndex (1.3.6.1.2.1.4.22.1.1)

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

ipNetToMediaPhysAddress (1.3.6.1.2.1.4.22.1.2)

The media-dependent `physical' address.

Syntax PhysAddress

Access read-write

Status mandatory

Return Value Writes not supported.

ipNetToMediaNetAddress (1.3.6.1.2.1.4.22.1.3)

The IpAddress corresponding to the media-dependent `physical' address.

Syntax IpAddress

Access read-write

Status mandatory

Return Value Writes not supported.

ipNetToMediaType (1.3.6.1.2.1.4.22.1.4)

The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively disassociates the interface identified with the entry from the mapping identified with the entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

Additional IP Objects

Following are the additional IP objects.

ipRoutingDiscards (1.3.6.1.2.1.4.23)

The number of routing entries that were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries.

Syntax Counter

Access read-only

Status mandatory

The ICMP group

Implementation of the ICMP group is mandatory for all systems.

icmpInMsgs (1.3.6.1.2.1.5.1)

The total number of ICMP messages received by the entity. This counter includes all those counted by icmpInErrors.

Syntax Counter

Access read-only

Status mandatory

icmpInErrors (1.3.6.1.2.1.5.2)

The number of ICMP messages received by the entity but were determined as having ICMP-specific errors (such as, bad ICMP checksums, bad length).

Syntax Counter

Access read-only

Status mandatory

icmpInDestUnreachs (1.3.6.1.2.1.5.3)

The number of ICMP Destination Unreachable messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInTimeExcds (1.3.6.1.2.1.5.4)

The number of ICMP Time Exceeded messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInParmProbs (1.3.6.1.2.1.5.5)

The number of ICMP Parameter Problem messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInSrcQuenchs (1.3.6.1.2.1.5.6)

The number of ICMP Source Quench messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInRedirects (1.3.6.1.2.1.5.7)

The number of ICMP Redirect messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInEchos (1.3.6.1.2.1.5.8)

The number of ICMP Echo (request) messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInEchoReps (1.3.6.1.2.1.5.9)

The number of ICMP Echo Reply messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInTimestamps (1.3.6.1.2.1.5.10)

The number of ICMP Timestamp (request) messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInTimestampReps (1.3.6.1.2.1.5.11)

The number of ICMP Timestamp Reply messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInAddrMasks (1.3.6.1.2.1.5.12)

The number of ICMP Address Mask Request messages received.

Syntax Counter

Access read-only

Status mandatory

icmpInAddrMaskReps (1.3.6.1.2.1.5.13)

The number of ICMP Address Mask Reply messages received.

Syntax Counter

Access read-only

Status mandatory

icmpOutMsgs (1.3.6.1.2.1.5.14)

The total number of ICMP messages that this entity attempted to send. This counter includes all those counted by icmpOutErrors.

Syntax Counter

Access read-only

Status mandatory

icmpOutErrors (1.3.6.1.2.1.5.15)

The number of ICMP messages that this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations, there may be no types of errors that contribute to this counter's value.

Syntax Counter

Access read-only

Status mandatory

icmpOutDestUnreachs (1.3.6.1.2.1.5.16)

The number of ICMP Destination Unreachable messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutTimeExcds (1.3.6.1.2.1.5.17)

The number of ICMP Time Exceeded messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutParmProbs (1.3.6.1.2.1.5.18)

The number of ICMP Parameter Problem messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutSrcQuenchs (1.3.6.1.2.1.5.19)

The number of ICMP Source Quench messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutRedirects (1.3.6.1.2.1.5.20)

The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.

Syntax Counter

Access read-only

Status mandatory

icmpOutEchos (1.3.6.1.2.1.5.21)

The number of ICMP Echo (request) messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutEchoReps (1.3.6.1.2.1.5.22)

The number of ICMP Echo Reply messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutTimestamps (1.3.6.1.2.1.5.23)

The number of ICMP Timestamp (request) messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutTimestampReps (1.3.6.1.2.1.5.24)

The number of ICMP Timestamp Reply messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutAddrMasks (1.3.6.1.2.1.5.25)

The number of ICMP Address Mask Request messages sent.

Syntax Counter

Access read-only

Status mandatory

icmpOutAddrMaskReps (1.3.6.1.2.1.5.26)

The number of ICMP Address Mask Reply messages sent.

Syntax Counter

Access read-only

Status mandatory

The TCP group

Implementation of the TCP group is mandatory for all systems that implement the TCP. Instances of object types that represent information about a particular TCP connection are transient; they persist only as long as the connection in question.

tcpRtoAlgorithm (1.3.6.1.2.1.6.1)

The algorithm used to determine the timeout value used for retransmitting unacknowledged octets.

Syntax INTEGER

Access read-only

Status mandatory

tcpRtoMin (1.3.6.1.2.1.6.2)

The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.

Syntax INTEGER

Access read-only

Status mandatory

tcpRtoMax (1.3.6.1.2.1.6.3)

The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC 793.

Syntax INTEGER

Access read-only

Status mandatory

tcpMaxConn (1.3.6.1.2.1.6.4)

The limit on the total number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value -1.

Syntax INTEGER

Access read-only

Status mandatory

tcpActiveOpens (1.3.6.1.2.1.6.5)

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

Syntax Counter

Access read-only

Status mandatory

tcpPassiveOpens (1.3.6.1.2.1.6.6)

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

Syntax Counter

Access read-only

Status mandatory

tcpAttemptFails (1.3.6.1.2.1.6.7)

The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

Syntax Counter

Access read-only

Status mandatory

tcpEstabResets (1.3.6.1.2.1.6.8)

The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

Syntax Counter

Access read-only

Status mandatory

tcpCurrEstab (1.3.6.1.2.1.6.9)

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

Syntax Gauge

Access read-only

Status mandatory

tcpInSegs (1.3.6.1.2.1.6.10)

The total number of segments received, including those received in error. This count includes segments received on currently established connections.

Syntax Counter

Access read-only

Status mandatory

tcpOutSegs (1.3.6.1.2.1.6.11)

The total number of segments sent including those on current connections, but excluding those containing only retransmitted octets.

Syntax Counter

Access read-only

Status mandatory

tcpRetransSegs (1.3.6.1.2.1.6.12)

The total number of segments retransmitted. That is, the number of TCP segments transmitted containing one or more previously transmitted octets.

Syntax Counter

Access read-only

Status mandatory

The TCP connection table

The TCP connection table contains information about this entity's existing TCP connections.

tcpConnState (1.3.6.1.2.1.6.13.1.1)

The state of this TCP connection. The only value that may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a "badValue" response if a management station attempts to set this object to any other value.

If a management station sets this object to the value deleteTCB(12), then this has the effect of deleting the TCB (as defined in RFC 793) of the corresponding connection on the managed Node. The result is an immediate termination of the connection.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Writes not supported.

tcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)

The local IP address for this TCP connection. In the case of a connection in the listen state that is willing to accept connections for any IP interface associated with the Node, the value 0.0.0.0 is used.

Syntax IpAddress

Access read-only

Status mandatory

tcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)

The local port number for this TCP connection.

Syntax INTEGER (0..65535)

Access read-only

Status mandatory

tcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)

The remote IP address for this TCP connection.

Syntax IpAddress

Access read-only

Status mandatory

tcpConnRemPort (1.3.6.1.2.1.6.13.1.5)

The remote port number for this TCP connection.

Syntax INTEGER (0..65535)

Access read-only

Status mandatory

Additional TCP objects

Following are the additional TCP objects.

tcpInErrs (1.3.6.1.2.1.6.14)

The total number of segments received in error (for example, bad TCP checksums).

Syntax Counter

Access read-only

Status mandatory

tcpOutRsts (1.3.6.1.2.1.6.15)

The number of TCP segments sent containing the RST flag.

Syntax Counter

Access read-only

Status mandatory

The UDP group

Implementation of the UDP group is mandatory for all systems that implement the UDP.

udpInDatagrams (1.3.6.1.2.1.7.1)

The total number of UDP datagrams delivered to UDP users.

Syntax Counter

Access read-only

Status mandatory

udpNoPorts (1.3.6.1.2.1.7.2)

The total number of received UDP datagrams for which there was no application at the destination port.

Syntax Counter

Access read-only

Status mandatory

udpInErrors (1.3.6.1.2.1.7.3)

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Syntax Counter

Access read-only

Status mandatory

udpOutDatagrams (1.3.6.1.2.1.7.4)

The total number of UDP datagrams sent from this entity.

Syntax Counter

Access read-only

Status mandatory

The UDP listener table

The UDP listener table contains information about this entity's UDP end-points on which a local application is currently accepting datagrams.

udpLocalAddress (1.3.6.1.2.1.7.5.1.1)

The local IP address for this UDP listener. In the case of a UDP listener that is willing to accept datagrams for any IP interface associated with the Node, the value 0.0.0.0 is used.

Syntax IpAddress

Access read-only

Status mandatory

udpLocalPort (1.3.6.1.2.1.7.5.1.2)

The local port number for this UDP listener.

Syntax INTEGER (0..65535)

Access read-only

Status mandatory

The EGP Group

Implementation of the EGP group is mandatory for all systems that implement the EGP.

egpInMsgs (1.3.6.1.2.1.8.1)

The number of EGP messages received without error.

Syntax Counter

Access read-only

Status mandatory

egpInErrors (1.3.6.1.2.1.8.2)

The number of EGP messages received that proved to be in error.

Syntax Counter

Access read-only

Status mandatory

egpOutMsgs (1.3.6.1.2.1.8.3)

The total number of locally generated EGP messages.

Syntax Counter

Access read-only

Status mandatory

egpOutErrors (1.3.6.1.2.1.8.4)

The number of locally generated EGP messages not sent due to resource limitations within an EGP entity.

Syntax Counter

Access read-only

Status mandatory

The EGP neighbor table

The EGP neighbor table contains information about this entity's EGP neighbors.

egpNeighState (1.3.6.1.2.1.8.5.1.1)

The EGP state of the local system with respect to this entry's EGP neighbor. Each EGP state is represented by a value that is one greater than the numerical value associated with the state in RFC 904.

Syntax INTEGER

Access read-only

Status mandatory

egpNeighAddr (1.3.6.1.2.1.8.5.1.2)

The IP address of this entry's EGP neighbor.

Syntax IpAddress

Access read-only

Status mandatory

egpNeighAs (1.3.6.1.2.1.8.5.1.3)

The autonomous system of this EGP peer. Zero should be specified if the autonomous system number of the neighbor is not yet known.

Syntax INTEGER

Access read-only

Status mandatory

egpNeighInMsgs (1.3.6.1.2.1.8.5.1.4)

The number of EGP messages received without error from this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighInErrs (1.3.6.1.2.1.8.5.1.5)

The number of EGP messages received from this EGP peer that proved to be in error (for example, bad EGP checksum).

Syntax Counter

Access read-only

Status mandatory

egpNeighOutMsgs (1.3.6.1.2.1.8.5.1.6)

The number of locally generated EGP messages to this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighOutErrs (1.3.6.1.2.1.8.5.1.7)

The number of locally generated EGP messages not sent to this EGP peer due to resource limitations within an EGP entity.

Syntax Counter

Access read-only

Status mandatory

egpNeighInErrMsgs (1.3.6.1.2.1.8.5.1.8)

The number of EGP-defined error messages received from this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighOutErrMsgs (1.3.6.1.2.1.8.5.1.9)

The number of EGP-defined error messages sent to this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighStateUps (1.3.6.1.2.1.8.5.1.10)

The number of EGP state transitions to the UP state with this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighStateDowns (1.3.6.1.2.1.8.5.1.11)

The number of EGP state transitions from the UP state to any other state with this EGP peer.

Syntax Counter

Access read-only

Status mandatory

egpNeighIntervalHello (1.3.6.1.2.1.8.5.1.12)

The interval between EGP Hello command retransmissions, in hundredths of a second. This represents the t1 timer as defined in RFC 904.

Syntax INTEGER

Access read-only

Status mandatory

egpNeighIntervalPoll (1.3.6.1.2.1.8.5.1.13)

The interval between EGP poll command retransmissions, in hundredths of a second. This represents the t3 timer as defined in RFC 904.

Syntax INTEGER

Access read-only

Status mandatory

egpNeighMode (1.3.6.1.2.1.8.5.1.14)

The polling mode of this EGP entity, either passive or active.

Syntax INTEGER { active(1), passive(2) }

Access read-only

Status mandatory

egpNeighEventTrigger (1.3.6.1.2.1.8.5.1.15)

A control variable used to trigger operator-initiated Start and Stop events. When read, this variable always returns the most recent value that egpNeighEventTrigger was set to. If it has not been set since the last initialization of the network management subsystem on the Node, it returns a value of “stop”.

When set, this variable causes a Start or Stop event on the specified neighbor, as specified on pages 8-10 of RFC 904. Briefly, a Start event causes an Idle peer to begin neighbor acquisition and a non-Idle peer to re-initiate neighbor acquisition. A stop event causes a non-Idle peer to return to the Idle state until a Start event occurs, either by egpNeighEventTrigger or otherwise.

Syntax INTEGER { start(1), stop(2) }

Access read-write

Status mandatory

egpAs (1.3.6.1.2.1.8.6)

The autonomous system number of this EGP entity.

Syntax INTEGER

Access read-only

Status mandatory

The transmission group

Based on the transmission media underlying each interface on a system, the corresponding portion of the Transmission group is mandatory for that system.

When Internet-standard definitions for managing transmission media are defined, the transmission group is used to provide a prefix for the names of those objects.

Typically, such definitions reside in the experimental portion of the MIB until they are proven, then as a part of the Internet standardization process, the definitions are accordingly elevated and a new object identifier, under the transmission group is defined. By convention, the name assigned is:

type OBJECT IDENTIFIER ::= { transmission number }.

Where "type" is the symbolic value used for the media in the ifType column of the ifTable object, and "number" is the actual integer value corresponding to the symbol.

The dot3StatTable

EtherLike-MIB:dot3StatsIndex (1.3.6.1.2.1.10.7.2.1.1)

An index value that uniquely identifies an interface to an ethernet-like medium. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Syntax Interger32

Access read-only

Status current

EtherLike-MIB:dot3StatsFCSErrors (1.3.6.1.2.1.10.7.2.1.3)

A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check. This count does not include frames received with frame-too-long or frame-too-short error.

Syntax Counter32

Access read-only

Status current

EtherLike-MIB:dot3StatsInternalMacReceiveErrors (1.3.6.1.2.1.10.7.2.1.16)

A count of frames for which reception on a particular interface fails due to an internal MAC sublayer receive error.

The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of receive errors on a particular interface that are not otherwise counted.

Syntax Counter32

Access read-only

Status current

EtherLike-MIB:dot3StatsSymbolErrors (1.3.6.1.2.1.10.7.2.1.18)

For an interface operating at 10 Gbps, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes an indication of a receive error.

Syntax Counter32

Access read-only

Status current

The dot3ControlTable

EtherLike-MIB:dot3ControlFunctionsSupported (1.3.6.1.2.1.10.7.9.1.1)

A list of the possible MAC Control functions implemented for this interface.

Syntax Bits

pause (0)

Access read-only

Status current

The dot3PauseTable

EtherLike-MIB:dot3PauseAdminMode (1.3.6.1.2.1.10.7.10.1.1)

This object represents the administratively configured PAUSE mode for this interface.

- disabled (1)
- enabledXmit (2)
- enabledRcv (3)
- enabledXmitAndRcv (4)

Syntax integer

Access read-only

Status current

EtherLike-MIB:dot3PauseOperMode (1.3.6.1.2.1.10.7.10.1.2)

This object reflects the PAUSE mode currently in use on this interface.

- disabled (1)
- enabledXmit (2)
- enabledRcv (3)
- enabledXmitAndRcv (4)

Syntax integer

Access read-only

Status current

EtherLike-MIB:dot3InPauseFrames (1.3.6.1.2.1.10.7.10.1.3)

A count of MAC Control frames received on this interface with an opcode indicating the PAUSE operation.

Syntax counter32

Access read-only

Status current

EtherLike-MIB:dot3OutPauseFrames (1.3.6.1.2.1.10.7.10.1.4)

A count of MAC Control frames transmitted on this interface with an opcode indicating the PAUSE operation.

Syntax counter32

Access read-only

Status current

The SNMP group

Implementation of the SNMP group is mandatory for all systems that support an SNMP protocol entity. Some of the objects defined below will be zero-valued in those SNMP implementations that are optimized to support only those functions specific to either a management agent or a management station. In particular, it should be observed that the objects below refer to an SNMP entity, and there may be several SNMP entities residing on a managed Node.

snmpInPkts (1.3.6.1.2.1.11.1)

The total number of messages delivered to the SNMP entity from the transport service.

Syntax Counter

Access read-only

Status mandatory

snmpOutPkts (1.3.6.1.2.1.11.2)

The total number of SNMP messages passed from the SNMP protocol entity to the transport service.

Syntax Counter

Access read-only

Status mandatory

snmpInBadVersions (1.3.6.1.2.1.11.3)

The total number of SNMP messages delivered to the SNMP protocol entity and were for an unsupported SNMP version.

Syntax Counter

Access read-only

Status mandatory

snmpInBadCommunityNames (1.3.6.1.2.1.11.4)

The total number of SNMP messages delivered to the SNMP protocol entity that used a SNMP community name not known to the entity.

Syntax Counter

Access read-only

Status mandatory

snmpInBadCommunityUses (1.3.6.1.2.1.11.5)

The total number of SNMP messages delivered to the SNMP protocol entity that represented an SNMP operation that was not allowed by the SNMP community named in the message.

Syntax Counter

Access read-only

Status mandatory

snmpInASNParseErrs (1.3.6.1.2.1.11.6)

The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP messages.

Syntax Counter

Access read-only

Status mandatory

snmpInTooBigs (1.3.6.1.2.1.11.8)

The total number of SNMP PDUs delivered to the SNMP protocol entity and for which the value of the error-status field is “tooBig”.

Syntax Counter

Access read-only

Status mandatory

snmpInNoSuchNames (1.3.6.1.2.1.11.9)

The total number of SNMP PDUs delivered to the SNMP protocol entity and for which the value of the error-status field is `NoSuchName`.

Syntax Counter

Access read-only

Status mandatory

snmpInBadValues (1.3.6.1.2.1.11.10)

The total number of SNMP PDUs delivered to the SNMP protocol entity and for which the value of the error-status field is “badValue”.

Syntax Counter

Access read-only

Status mandatory

snmpInReadOnlys (1.3.6.1.2.1.11.11)

The total number valid SNMP PDUs delivered to the SNMP protocol entity and for which the value of the error-status field is “readOnly”. It should be noted that it is a protocol error to generate an SNMP PDU that contains the value `readOnly` in the error-status field, as such, this object is provided as a means of detecting incorrect implementations of the SNMP.

Syntax Counter

Access read-only

Status mandatory

snmpInGenErrs (1.3.6.1.2.1.11.12)

The total number of SNMP PDUs delivered to the SNMP protocol entity and for which the value of the error-status field is “genErr”.

Syntax Counter

Access read-only

Status mandatory

snmpInTotalReqVars (1.3.6.1.2.1.11.13)

The total number of MIB objects retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.

Syntax Counter

Access read-only

Status mandatory

snmpInTotalSetVars (1.3.6.1.2.1.11.14)

The total number of MIB objects altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.

Syntax Counter

Access read-only

Status mandatory

snmpInGetRequests (1.3.6.1.2.1.11.15)

The total number of SNMP Get-Request PDUs accepted and processed by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpInGetNexts (1.3.6.1.2.1.11.16)

The total number of SNMP Get-Next PDUs accepted and processed by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpInSetRequests (1.3.6.1.2.1.11.17)

The total number of SNMP Set-Request PDUs accepted and processed by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpInGetResponses (1.3.6.1.2.1.11.18)

The total number of SNMP Get-Response PDUs accepted and processed by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpInTraps (1.3.6.1.2.1.11.19)

The total number of SNMP Trap PDUs accepted and processed by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpOutTooBigs (1.3.6.1.2.1.11.20)

The total number of SNMP PDUs generated by the SNMP protocol entity and for which the value of the error-status field is “tooBig”'

Syntax Counter

Access read-only

Status mandatory

snmpOutNoSuchNames (1.3.6.1.2.1.11.21)

The total number of SNMP PDUs generated by the SNMP protocol entity and for which the value of the error-status is `NoSuchName`.

Syntax Counter

Access read-only

Status mandatory

snmpOutBadValues (1.3.6.1.2.1.11.22)

The total number of SNMP PDUs generated by the SNMP protocol entity and for which the value of the error-status field is “badValue”.

Syntax Counter

Access read-only

Status mandatory

snmpOutGenErrs (1.3.6.1.2.1.11.24)

The total number of SNMP PDUs generated by the SNMP protocol entity and for which the value of the error-status field is “genErr”.

Syntax Counter

Access read-only

Status mandatory

snmpOutGetRequests (1.3.6.1.2.1.11.25)

The total number of SNMP Get-Request PDUs generated by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpOutGetNexts (1.3.6.1.2.1.11.26)

The total number of SNMP Get-Next PDUs generated by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpOutSetRequests (1.3.6.1.2.1.11.27)

The total number of SNMP Set-Request PDUs generated by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpOutGetResponses (1.3.6.1.2.1.11.28)

The total number of SNMP Get-Response PDUs generated by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpOutTraps (1.3.6.1.2.1.11.29)

The total number of SNMP Trap PDUs generated by the SNMP protocol entity.

Syntax Counter

Access read-only

Status mandatory

snmpEnableAuthenTraps (1.3.6.1.2.1.11.30)

Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authentication-failure traps may be disabled.

It is strongly recommended that this object be stored in non-volatile memory so that it remains constant between re-initializations of the network management system.

Syntax INTEGER { enabled(1), disabled(2) }

Access read-write

Status mandatory

Return Value Read returns enabled (1) if AuthFailureTrap = True, otherwise disabled (2). Writes not supported.

The ifXTable

ifName (1.3.6.1.2.1.31.1.1.1.1)

The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device. If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string.

Syntax DisplayString

Access read-only

Status current

ifInMulticastPkts (1.3.6.1.2.1.31.1.1.1.2)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses.

Syntax Counter32

Access read-only

Status current

ifInBroadcastPkts (1.3.6.1.2.1.31.1.1.1.3)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were addressed to a broadcast address at this sub-layer.

Syntax Counter32

Access read-only

Status current

ifOutMulticastPkts (1.3.6.1.2.1.31.1.1.1.4)

The total number of packets that higher-level protocols requested be transmitted, and that were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses.

Syntax Counter32

Access read-only

Status current

ifOutBroadcastPkts (1.3.6.1.2.1.31.1.1.1.5)

The total number of packets that higher-level protocols requested be transmitted, and that were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent.

Syntax Counter32

Access read-only

Status current

ifHighSpeed (1.3.6.1.2.1.31.1.1.1.15)

An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer that has no concept of bandwidth, this object should be zero.

Syntax Gauge32

Access read-only

Status current

ifPromiscuousMode (1.3.6.1.2.1.31.1.1.1.16)

This object has a value of false(2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets/frames transmitted on the media.

- true(1)
- false(2)

Syntax TruthValue

Access read-only

Status current

ifConnectorPresent (1.3.6.1.2.1.31.1.1.1.17)

This object has the value 'true(1)' if the interface sublayer has a physical connector and the value 'false(2)' otherwise.

- true(1)
- false(2)

Syntax TruthValue

Access read-only

Status current

ifAlias (1.3.6.1.2.1.31.1.1.1.18)

This object is an 'alias' name for the interface as specified by a network manager, and provides a non-volatile 'handle' for the interface.

Syntax DisplayString

Access read-only

Status current

ifCounterDiscontinuityTime (1.3.6.1.2.1.31.1.1.1.19)

The value of sysUpTime on the most recent occasion that any one or more of this interface's counters suffered a discontinuity. The relevant counters are the specific instances associated with this interface of any Counter32 or Counter64 object contained in the ifTable or ifXTable. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value.

Syntax TimeStamp

Access read-only

Status current

ifTableLastChange (1.3.6.1.2.1.31.5)

The value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable. If the number of entries has been unchanged since the last re-initialization of the local network management subsystem, then this object contains a zero value.

Syntax TimeTicks

Access read-only

Status current

Chapter 5. Fibre Alliance MIB objects

This chapter covers the implementation details for the Fibre Alliance Management Information Bases (FA-MIB) version 6.0 on the IBM FC3171 switch.

FA MIB definitions

The FA-MIB version 4.0 is a collection of structured objects that resides on the workstation with the manager application. These objects define the syntax for information exchanged between the manager and the agent. The textual substitutions in Table 3 are specific to the FA-MIB and can be used in place of primitive data types.

Table 3. FA-MIB textual substitutions

Description	Syntax
FcNameId	OCTET STRING (SIZE(8))
FcGlobalId	OCTET STRING (SIZE(16))
FcAddressId	OCTET STRING (SIZE(3))

FcEventSeverity	INTEGER {
	unknown (1),
	emergency (2),
	alert (3),
	critical (4),
	error (5),
	warning (6),
	notify (7),
	info (8),
	debug (9),
	mark (10) - All messages logged
	}

Table 3. FA-MIB textual substitutions (Continued)

Description	Syntax

FcUnitType	INTEGER {
	unknown(1)
	other(2)—none of the following
	hub(3)—passive connectivity unit supporting loop protocol.
	switch(4)—active connectivity unit supporting multiple protocols.
	gateway(5)—unit that converts not only the interface but also encapsulates the frame into another protocol. The assumption is that there is always two gateways connected together. For example, FC <-> ATM.
	converter(6)—unit that converts from one interface to another. For example, FC <-> SCSI.
	hba(7)—host bus adapter
	proxy-agent(8)—software proxy-agent
	storage-device(9)—disk, cd, tape, etc.
	host(10)—host computer
	storage-subsystem(11)—raid, library, etc.
	module(12)—subcomponent of a system
	swdriver(13)—software driver
	storage-access-device(14)—Provides storage management and access for heterogeneous hosts and heterogeneous devices
	wdm(15)—waveform division multiplexer
	ups(16)—uninterruptable power supply
	}

revisionNumber

The revision number for this MIB. The format of the revision value is as follows:

- (0) = high order major revision number
- (1) = low order major revision number
- (2) = high order minor revision number
- (3) = low order minor revision number

The value will be stored as an ASCII value. The following is the current value of 04.00 for this object.

- (0) = '0'
- (1) = '4'
- (2) = '0'
- (3) = '0'

Syntax DisplayString (SIZE (4))

Access read-only

Status mandatory

Return Value A four digit ASCII value (for example, 0400 for MIB revision 4.0).

Connectivity unit group

The objects described in this section are not in a table format. An example of how to access one of these objects is:

```
"snmpget localhost public fcmgmt.connSet.uNumber.0".
```

uNumber (1.3.6.1.3.94.1.1)

The number of connectivity units present on this system (represented by this agent). May be a count of the boards in a chassis or the number of full boxes in a rack.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of switches in fabric

systemURL (1.3.6.1.3.94.1.2)

The top-level URL of the system. If it does not exist, the value is an empty string. The URL format is implementation dependant and can have keywords embedded that are preceded by a percent sign (for example, %USER).

Syntax DisplayString

Access read-write

Status mandatory

Return Value Switch IP address. For example, http://10.0.0.1. Writes not supported; returns `NoSuchName`.

statusChangeTime (1.3.6.1.3.94.1.3)

The sysUpTime timestamp at which the last status change occurred for any members of the set, in centiseconds.

Syntax TimeTicks

Access read only

Status obsolete

Return Value sysUpTime timestamp at which the last status change occurred

configurationChangeTime (1.3.6.1.3.94.1.4)

The sysUpTime timestamp at which the last configuration change occurred for any members of the set, in centiseconds. This represents a union of change information for connUnitConfigurationChangeTime.

Syntax TimeTicks

Access read only

Status obsolete

Return Value sysUpTime timestamp at which the last configuration change occurred

connUnitTableChangeTime (1.3.6.1.3.94.1.5)

The sysUpTime timestamp at which the connUnitTable was updated (an entry was either added or deleted), in centiseconds.

Syntax TimeTicks

Access read only

Status obsolete

Return Value sysUpTime timestamp at which the connUnitTable was updated

Connectivity table

The objects described in this section are in a table format indexed by switch World Wide Name. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitTable.connUnitEntry.connUnitId..16.0.0.192.221.0.
144.167.0.0.0.0.0.0.0".
```

connUnitId (1.3.6.1.3.94.1.6.1.1)

The unique identification for this connectivity unit among those within this proxy domain. The value must be unique within the proxy domain because it is the index variable for connUnitTable. The value assigned to a given connectivity unit should be persistent across agent and unit resets. It should be the same as connUnitGlobalId if connUnitGlobalId is known and stable.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitGlobalId (1.3.6.1.3.94.1.6.1.2)

An optional global-scope identifier for this connectivity unit. It must be a WWN for this connectivity unit or 16 octets of value zero.

Syntax connUnitGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitType (1.3.6.1.3.94.1.6.1.3)

The type of this connectivity unit.

Syntax FcUnitType

Access read-only

Status mandatory

Return Value switch (4)

connUnitNumports (1.3.6.1.3.94.1.6.1.4)

Number of physical ports in the connectivity unit (internal/embedded, external).

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of ports on the switch

connUnitState (1.3.6.1.3.94.1.6.1.5)

The operational state of the switch mapped. The overall state of connectivity unit.

Syntax

INTEGER

Access

read-only

Status

mandatory

Return Value

Table 4 lists the switch operational states.

Table 4. Switch operational states

Switch state	Return state
online	online (2)
offline	offline (3)
diagnostics	offline (3)
other	unknown (1)

connUnitStatus (1.3.6.1.3.94.1.6.1.6)

Overall status of the connectivity unit. The goal of this object is to be the single poll point to check the status of the connunit. If there is any other component that has warning, then this should be set to warning. Any of these values may occur with any of the ConnUnitState values.

Syntax

INTEGER

Access

read-only

Status

mandatory

Return Value

`OK (3)` unless one or more of the events listed Table 5 occurs

Table 5. Connectivity unit return values

Event	Return Value
One or more cooling fans fail	warning (4)
All cooling fans fail	failed (5)
Temperature status = Warm	warning (4)
Temperature status = Overheating	failed (5)
Any port down	warning (4)
Switch is offline or in diagnostics mode	warning (4)
Switch is down	failed (5)

connUnitProduct (1.3.6.1.3.94.1.6.1.7)

The sml attribute Config.Snmp.SysDescr. This is the system description shown on the 'show version' telnet screen. It can also be read on the 'show setup snmp' screen and written using the 'set setup snmp' Telnet screen.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value `IBM Flex System FC3171 8 Gb SAN Switch` or `IBM Flex System FC3171 8 Gb Pass-thru`.

connUnitSn (1.3.6.1.3.94.1.6.1.8)

The serial number for this connectivity unit.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Chassis serial number.

connUnitUpTime (1.3.6.1.3.94.1.6.1.9)

The number of centiseconds since the last unit initialization.

Syntax TimeTicks

Access read-only

Status mandatory

Return Value Time interval since either POST or a reset (not including Hotreset command for the NDCLA feature). POST (Power-On Self-Test) occurs during Power-On, or hardreset.

connUnitUrl (1.3.6.1.3.94.1.6.1.10)

URL to launch a management application, if applicable. Otherwise, it is an empty string. In a standalone unit, this would be the same as the top-level URL. This has the same definition as systemURL for keywords. If write is not supported, then the return value is invalid. This value will be retained across boots.

Syntax DisplayString

Access read-write

Status mandatory

Return Value Switch IP address. For example, http://10.0.0.1. Writes not supported, returns `NoSuchName`.

connUnitDomainId (1.3.6.1.3.94.1.6.1.11

24 bit Fibre Channel address ID of this connectivity unit, right justified with leading zeros if required. This should be set to the Fibre Channel address ID, or if it is a switch, it would be set to the Domain Controller address. If this value is not applicable, return all bits set to one.

Syntax OCTET STRING (SIZE(3))

Access read-only

Status mandatory

Return Value Domain controller address, for example, FF FC 65.

connUnitProxyMaster (1.3.6.1.3.94.1.6.1.12)

A value of “yes” means this is the proxy master unit for a set of managed units. For example, this could be the only unit with a management card in it for a set of units. A standalone unit should return “yes” for this object.

Syntax

INTEGER {
unknown(1),
no(2),
yes(3)
}

Access

read-only

Status

mandatory

Return Value

Returns `yes (3)` for an out-of-band switch; returns `no (2)` for an in-band switch

connUnitPrincipal (1.3.6.1.3.94.1.6.1.13)

Whether this connectivity unit is the principal unit within the group of fabric elements. If this value is not applicable, the return is unknown.

Syntax

INTEGER {
unknown(1),
no(2),
yes(3)
}

Access

read-only

Status

mandatory

Return Value

Returns `yes (3)` for the principal switch, otherwise returns `no (2)`

connUnitNumSensors (1.3.6.1.3.94.1.6.1.14)

Number of sensors in the connUnitSensorTable elements. If this value is not applicable, return unknown.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 8

connUnitStatusChangeTime (1.3.6.1.3.94.1.6.1.15)

The sysUpTime timestamp, in centiseconds, at which the last status change occurred.

Syntax TimeTicks

Access read-only

Status obsolete

Return Value This object is obsolete. Always returns error status `NoSuchName`.

connUnitConfigurationChangeTime (1.3.6.1.3.94.1.6.1.16)

The sysUpTime timestamp, in centiseconds, at which the last configuration change occurred.

Syntax TimeTicks

Access read-only

Status obsolete

Return Value This object is obsolete. Always returns error status `NoSuchName`.

connUnitNumRevs (1.3.6.1.3.94.1.6.1.17)

The number of revisions in the connUnitRevsTable.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 3 (The revision number for all switch components.)

connUnitNumZones (1.3.6.1.3.94.1.6.1.18)

Number of zones defined in connUnitZoneTable.

Syntax INTEGER

Access read-only

Status obsolete

Return Value This object is obsolete. Always returns error status `NoSuchName`.

connUnitModuleId (1.3.6.1.3.94.1.6.1.19)

This is a unique ID, persistent between boots, that can be used to group a set of connUnits together into a module. The intended use would be to create a connUnit with a connUnitType of “module” to represent a physical or logical group of connectivity units. Then, the value of the group would be set to the value of connUnitId for this “container” connUnit. connUnitModuleId should be zeros if this connUnit is not part of a module.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitName (1.3.6.1.3.94.1.6.1.20)

A display string containing a name for this connectivity unit. This object value should be persistent between boots.

Syntax DisplayString (SIZE(0..79))

Access read-write

Status mandatory

Return Value `IBM8Gb`

connUnitInfo (1.3.6.1.3.94.1.6.1.21)

A display string containing information about this connectivity unit. This object value should be persistent between boots.

Syntax DisplayString

Access read-write

Status mandatory

Return Value ConfigDescription field for the switch. Default: `Default Config`

connUnitControl (1.3.6.1.3.94.1.6.1.22)

This object is used to control the addressed connUnit. “Cold Start” and “Warm Start” are as defined in MIB-II and are not meant to be a factory reset.

- resetConnUnitColdStart: the addressed unit performs a “Cold Start” reset.
- resetConnUnitWarmStart: the addressed unit performs a “Warm Start” reset.
- offlineConnUnit: the addressed unit puts itself into an implementation dependant “offline” state. In general, if a unit is in an offline state, it cannot be used to perform meaningful Fibre Channel work.
- onlineConnUnit: the addressed unit puts itself into an implementation dependant “online” state. In general, if a unit is in an online state, it is capable of performing meaningful Fibre Channel work.

Each implementation may chose not to allow any or all of these values on a SET.

Syntax

INTEGER {
unknown(1),
invalid(2),
resetConnUnitColdStart(3),
resetConnUnitWarmStart(4),
offlineConnUnit(5),
onlineConnUnit(6)
}

Access

read-write

Status

mandatory

Return Value

The following tables list the connUnitControl read return values and the connUnitControl write control values.

Table 6. connUnitContol read return values

Switch setting	Return value
Online	Online (6)
Offline	Offline (5)
Diagnostics	Offline (5)
Other	Unknown (1)

Table 7. connUnitContol write control values

Control value	Result
Cold Reset (3)	Reset
Offline (5)	Offline
Online (6)	Online
other	Not supported

connUnitContact (1.3.6.1.3.94.1.6.1.23)

Contact information for this connectivity unit, and is persistent across boots.

Syntax DisplayString (SIZE (0..79))

Access read-write

Status mandatory

Return Value Default: `<sysContact undefined>`. The string size is limited to a maximum of 64.

connUnitLocation (1.3.6.1.3.94.1.6.1.24)

Location information for this connectivity unit, and is persistent across boots.

Syntax DisplayString (SIZE (0..79))

Access read-write

Status mandatory

Return Value Default: `<sysLocation undefined>`. The string size is limited to a maximum of 64.

connUnitEventFilter (1.3.6.1.3.94.1.6.1.25)

This value defines the event severity that will be logged by this connectivity unit. All events of severity less than or equal to connUnitEventFilter are logged in connUnitEventTable.

Syntax

FcEventSeverity

Access

read-write

Status

mandatory

Return Value

The following tables list the connUnitEventFilter read return values and control write values.

Table 8. connUnitEventFilter read return values

Severity levels	Return value
Critical	Critical (4)
Warn	Warning (6)
Info	Info (8)
None	Unknown (1)

Table 9. connUnitEventFilter control write values

Control value	Result
Emergency (2)	Critical
Alert (3)	Critical
Critical (4)	Critical
Error (5)	Warn
Warning (6)	Warn
Notify (7)	Info
Info (8)	Info
Debug (9)	Info
Mark (10)	Info
Unknown (1)	None

connUnitNumEvents (1.3.6.1.3.94.1.6.1.26)

Number of events currently in the connUnitEventTable.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of events in the event table

connUnitMaxEvents (1.3.6.1.3.94.1.6.1.27)

Maximum number of events that can be defined in connUnitEventTable.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 30

connUnitEventCurrID (1.3.6.1.3.94.1.6.1.28)

The last used event ID (connUnitEventIndex).

Syntax INTEGER

Access read-only

Status mandatory

Return Value Event ID of the last event

connUnitFabricID (1.3.6.1.3.94.1.6.1.29)

A globally unique value to identify the fabric that this ConnUnit belongs to, otherwise empty string. This would typically be equal to the connUnitGlobalID of the primary switch in a Fibre Channel fabric.

Syntax FcGlobalId

MaxAccess read-only

Status mandatory

Return Value World wide name of the principal switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitNumLinks (1.3.6.1.3.94.1.6.1.30)

The number of links in the link table.

Syntax INTEGER

MaxAccess read-only

Status mandatory

Return Value Number of link table entries for each switch

connUnitVendorId (1.3.6.1.3.94.1.6.1.31)

The connectivity unit vendor's name.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value `IBM`

Revision table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Table of revisions for hardware and software elements. There are four revision items in each switch. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitRevsTable.connUnitRevsEntry.connUnitRevsUnitId.16
.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

The number of entries in this table varies depending on the platform that is being examined and the number of blades installed. SNMP first reports the firmware revision and flasher shell version. It then iterates through each of the installed blades reporting the PCB revision and ASIC version.

connUnitRevsUnitId (1.3.6.1.3.94.1.7.1.1)

The connUnitId of the connectivity unit that contains this revision table.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitRevsIndex (1.3.6.1.3.94.1.7.1.2)

A unique value among all connUnitRevsEntrys with the same value of connUnitRevsUnitId, in the range between 1 and connUnitNumRevs[connUnitRevsUnitId].

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Revision table index

connUnitRevsRevId (1.3.6.1.3.94.1.7.1.3)

A vendor-specific string identifying a revision of a component of the connUnit indexed by connUnitRevsUnitId.

Syntax

DisplayString

Access

read-only

Status

mandatory

Return Value

Table 10 lists the connUnitRevsRevtId return values.

Table 10. ConnUnitRevsRevId return values

Table index	Return value
1	Active Firmware Image
2	Flasher Shell Version
3	Hardware ASIC Version (1 per blade)

connUnitRevsDescription (1.3.6.1.3.94.1.7.1.4)

Description of a component to which the revision corresponds.

Syntax DisplayString

Access read-only

Status mandatory

Return Value Table 11 lists the connUnitRevsDescription return values.

Table 11. ConnUnitRevsDescription return values

Table index	Return value
1	Active Firmware Version
2	Flasher Shell Version
3	Hardware ASIC Version

Sensor table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Index is the sensor number being interrogated. There are six sensor items in each switch. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitSensorTable.connUnitSensorEntry.connUnitSensorUni
tId.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

connUnitSensorUnitId (1.3.6.1.3.94.1.8.1.1)

The connUnitId of the connectivity unit that contains this sensor table.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitSensorIndex (1.3.6.1.3.94.1.8.1.2)

A unique value among all connUnitSensorEntrys with the same value of connUnitSensorUnitId, in the range between 1 and connUnitNumSensor[connUnitSensorUnitId].

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Sensor table index

connUnitSensorName (1.3.6.1.3.94.1.8.1.3)

A textual identification of the sensor intended primarily for operator use.

Syntax DisplayString

Access read-only

Status mandatory

Return Value Table 12 lists the connUnitSensorName return values.

Table 12. ConnUnitSensorName return values

Table index	Return value
1	Power supply 1 status
2	Temperature status
3	Temperature sensor 1 value
4	Temperature sensor 2 value
5	Temperature sensor 3 value
6	Temperature sensor 4 value
7	Temperature sensor 5 value
8	Temperature sensor 6 value

connUnitSensorStatus (1.3.6.1.3.94.1.8.1.4)

The status indicated by the sensor.

Syntax

INTEGER {
unknown(1)
other(2) - the sensor indicates other than ok (warning or failure).
ok(3) - the sensor indicates ok
warning(4) - the sensor indicates a warning
failed(5) - the sensor indicates failure
}

Access

read-only

Status

mandatory

Return Value

Table 13 lists the connUnitSensorStatus board temperature return values.

Table 13. ConnUnitSensorStatus board temperature return values

Switch Value	Return value
Normal	OK (3)
Warm	Warning (4)
Overheating	Failed (5)
Other	Unknown (1)

Table 14 lists the connUnitSensorStatus fan status return values.

Table 14. ConnUnitSensorStatus fan status return values

Switch value	Return value
Good	OK (3)
Bad	Failed (5)
Other	Unknown (1)

Table 15 lists the connUnitSensorStatus board temperature return values.

Table 15. ConnUnitSensorStatus voltage status return values

Switch value	Return value
Good	OK (3)
Bad	Failed (5)
Other	Unknown (1)

connUnitSensorInfo (1.3.6.1.3.94.1.8.1.5)

Miscellaneous static information about the sensor, such as its serial number.

Syntax DisplayString

Access read-only

Status mandatory

Return Value Empty string

connUnitSensorMessage (1.3.6.1.3.94.1.8.1.6)

This describes the status of the sensor as a message. It may also provide more resolution on the sensor indication. For example, `Cover temperature 1503K, above nominal operating range` ::= { connUnitSensorEntry 6 }.

Syntax

DisplayString

Access

read-only

Status

mandatory

Return Value

Table 16 lists the connUnitSensorMessage values.

Table 16. ConnUnitSensorMessage values

Sensor	Value
Power supply	Good/Bad/NotInstalled
Fan	Good/Bad/NotInstalled
Temperature status	Normal/Warm/Overheating/NotInstalled
Temperature value	Temperature °C

connUnitSensorType (1.3.6.1.3.94.1.8.1.7)

The type of component being monitored by this sensor.

Syntax

INTEGER {
unknown(1),
other(2),
battery(3),
fan(4),
power-supply(5),
transmitter(6),
enclosure(7),
board(8),
receiver(9)
}

Access

read-only

Status

mandatory

Return Value

Table 17 lists the connUnitSensorType return values.

Table 17. ConnUnitSensorType return values

Sensor	Value
Temperature	Board (8)
Fan	Fan (4)
Power supply	Power supply (5)
Voltage	Board (8)

connUnitSensorCharacteristic (1.3.6.1.3.94.1.8.1.8)

The characteristics being monitored by this sensor.

Syntax

INTEGER {
unknown(1),
other(2),
temperature(3),
pressure(4),
emf(5),
currentValue(6), - current is a keyword
airflow(7),
frequency(8),
power(9),
door(10)
}

Access

read-only

Status

mandatory

Return Value

Table 18 lists the connUnitSensorCharacterisitic values.

Table 18. ConnUnitSensorCharacteristic values

Sensor	Value
Temperature value	Temperature (3)
Temperature status	Temperature (3)
Fan	Airflow (7)
Power supply	Power (9)

Port table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Index is the port number being interrogated. There may be different numbers of ports in each switch so the agent must determine the maximum allowable index on a switch by switch basis. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitPortTable.connUnitPortEntry.connUnitPortUnitId.16
.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

connUnitPortUnitId (1.3.6.1.3.94.1.10.1.1)

The connUnitId of the connectivity unit that contains this port.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitPortIndex (1.3.6.1.3.94.1.10.1.2)

A unique value among all connUnitPortEntrys on this connectivity unit, between 1 and connUnitNumPort[connUnitPortUnitId].

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Index for each port on the switch: 1–20

connUnitPortType (1.3.6.1.3.94.1.10.1.3)

The port type.

Syntax

INTEGER {
unknown(1),
other(2),
not-present(3),
hub-port(4),
n-port(5), - end port for fabric
nl-port(6), - end port for loop
fl-port(7), - public loop
f-port(8), - fabric port
e-port(9), - fabric expansion port
g-port(10), - generic fabric port
domain-ctl(11), - domain controller
hub-controller(12),
scsi(13), - parallel SCSI port
escon(14),
lan(15),
wan(16),
ac(17), - AC power line
dc(18), - DC power line
ssa(19) - serial storage architecture
wdm(20),-- optical wave division multiplex
ib 21), - Infiniband
ipstore(22) - IP storage
}

Access

read-only

Status

mandatory

Return Value

Table 19 lists the connUnitPortType return values.

Table 19. ConnUnitPortType return values

Switch port type	Return value
G	g-port (10)
FL	fl-port (7)
F	f-port (8)
E	e-port (9)
Donor	other (2)
other	unknown (1)

connUnitPortFCClassCap (1.3.6.1.3.94.1.10.1.4)

Bit mask that specifies the classes of service capability of this port. If this is not applicable, return all bits set to zero.

The bits have the following definition:

unknown - 0

class-f - 1

class-one - 2

class-two - 4

class-three - 8

class-four - 16

class-five - 32

class-six - 64

Syntax OCTET STRING (SIZE (2))

Access read-only

Status mandatory

Return Value 0x0d (Class f, Class 2, and Class 3).

connUnitPortFCClassOp (1.3.6.1.3.94.1.10.1.5)

Bit mask that specifies the classes of service that are currently operational. If this is not applicable, return all bits set to zero. This object has the same definition as connUnitPortFCClassCap" ::= { connUnitPortEntry 5 }.

Syntax OCTET STRING (SIZE (2))

Access read-only

Status mandatory

Return Value If F or FL, returns 0x0c (Class 2, and Class 3), else returns 0x0d (Class f, Class 2, and Class 3).

connUnitPortState (1.3.6.1.3.94.1.10.1.6)

The user selected state of the port hardware.

Syntax

INTEGER {
unknown(1),
online(2), - available for meaningful work
offline(3), - not available for meaningful work
bypassed(4), - no longer used (4/12/00)
diagnostics(5)
}

Access

read-only

Status

mandatory

Return Value

Table 20 lists the connUnitPortState return values.

Table 20. ConnUnitPortState return values

Port value	Return value
Online	online (2)
Offline	offline (3)
Downed	offline (3)
Diagnostics	diagnostics (5)
other	unknown (1)

connUnitPortStatus (1.3.6.1.3.94.1.10.1.7)

An overall protocol status for the port. This value of connUnitPortState is not online, then this is reported Unknown.

Syntax

INTEGER {
unknown(1),
unused(2), - device cannot report this status
ready(3), - FCAL Loop or FCPH Link reset protocol; initialization complete
warning(4), - do not use (4/12/00)
failure(5), - do not use (4/12/00)
notparticipating(6), - loop not participating and does not have a loop address
initializing(7), - protocol is proceeding
bypass(8), - do not use (4/12/00)
ols(9) - FCP offline status
other(10) - status not described above
}

Access

read-only

Status

mandatory

Return Value

Unused (2)

connUnitPortTransmitterType (1.3.6.1.3.94.1.10.1.8)

The technology of the port transceiver.

Syntax

INTEGER {
unknown(1),
other(2),
unused(3),
shortwave(4),
longwave(5),
copper(6),
scsi(7),
longwaveNoOFC(8),
shortwaveNoOFC(9),
longwaveLED(10),
ssa(11)
{

Access

read-only

Status

mandatory

Return Value

Table 21 lists the connUnitPortTransmitterType return values.

Table 21. ConnUnitPortTransmitterType return values

SFP transmitter type	Return value
Not Installed	Unused (3)
SL	Shortwave (4)
LL	Longwave (5)
LC	LongwaveNoOFC (8)
SN	ShortwaveNoOFC (9)
EL	Copper (6)
Other	Unknown (1)

connUnitPortModuleType (1.3.6.1.3.94.1.10.1.9)

The module type of the port connector.

Syntax

INTEGER {
unknown(1),
other(2),
gbic(3),
embedded(4), - fixed (oneXnine)
glm(5),
gbicSerialId(6),
gbicNoSerialId(7),
gbicNotInstalled(8),
smallFormFactor(9) - this is generically a small form factor connector.
}

Access

read-only

Status

mandatory

Return Value

Table 22 lists the connUnitPortModuleType return values.

Table 22. ConnUnitPortModuleType return values

Type	Value
1 Gb/2Gb Ports	smallFormFactor(9)
10 Gb Ports	Other (2)

connUnitPortWwn (1.3.6.1.3.94.1.10.1.10)

The World Wide Name of the port, if applicable, otherwise returns all zeros.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value Port World Wide Name followed by 8 bytes of zeros. For example, the return value for port #2 would be 20 02 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00, and the return value for port #2 would be 20 0E 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00. If a port is configured as a Donor, return value = 0.

connUnitPortFCId (1.3.6.1.3.94.1.10.1.11)

This is the assigned Fibre Channel ID of this port. This value is expected to be a Big Endian value of 24 bits. If this is a loop, then it is the ALPA that is connected. If this is an E_Port, then it will only contain the domain ID left justified, zero filled. If this port does not have a Fibre Channel address, returns all bits set to 1.

Syntax FcAddressId

Access read-only

Status mandatory

Return Value Address for each port based on Domain, Area, and ALPA. For example, port #15 would be equal to 640F00 (Domain = 0x64, Area = 0x0F, ALPA = 0x00).

connUnitPortSn (1.3.6.1.3.94.1.10.1.12)

The serial number of the unit. If not applicable, returns an empty string.

Syntax DisplayString (SIZE(0..79))

Access read-only

Status unsupported

Return Value Media part number of the SFP (or MediaPartNumber) if installed.

connUnitPortRevision (1.3.6.1.3.94.1.10.1.13)

The port revision. For example, for a GBIC.

Syntax DisplayString (SIZE(0..79))

Access read-only

Status unsupported

Return Value Media revision of the SFP (or MediaRevision) if installed.

connUnitPortVendor (1.3.6.1.3.94.1.10.1.14)

The port vendor. For example, for a GBIC.

Syntax DisplayString (SIZE(0..79))

Access read-only

Status unsupported

Return Value Port vendor as reported by the SFP (if supported).

connUnitPortSpeed (1.3.6.1.3.94.1.10.1.15)

The speed of the port in kilobytes per second.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Operational speed, otherwise returns the administrative speed setting. If 1 Gbps, returns 106250. If 2 Gbps, returns 212500. If 4 Gbps, returns 425000. If 10 Gbps, returns 1062500.

connUnitPortControl (1.3.6.1.3.94.1.10.1.16)

This object is used to control the addressed connUnit's port.

- **resetConnUnitPort**: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific reset operation. Examples of these operations are: the Link Reset protocol, the Loop Initialization protocol, or a re-synchronization occurring between the transceiver in the addressed port to the transceiver that the port is connected to.
- **bypassConnUnitPort**: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "bypass" operation. Examples of these operations are transitioning from online to offline, a request (non-participating) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.
- **unbypassConnUnitPort**: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "unbypass" operation. Examples of these operations are the Link Failure protocol, a request (participating) command to the Loop Port state machine, or addition of the port to an arbitrated loop by a hub.
- **offlineConnUnitPort**: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "offline" operation. Examples of these operations are disabling a port's transceiver, the Link Failure protocol, request (non-participating) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.
- **onlineConnUnitPort**: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "online" operation. Examples of these operations are enabling a port's transceiver, the Link Failure protocol, request (participating) command to the Loop Port state machine, or addition of the port from an arbitrated loop by a hub.
- **resetConnUnitPortCounters**: If the addressed connUnit allows this operation to be performed to this port, the addressed port statistics table counters will be set to zero.

Each implementation may choose not to allow any or all of these values on a SET. On a read, if you do not support write, then return invalid. Otherwise, return the last control operation attempted.

Syntax

```
INTEGER {
unknown(1),
invalid(2),
resetConnUnitPort(3),
bypassConnUnitPort(4),
unbypassConnUnitPort(5),
offlineConnUnitPort(6),
onlineConnUnitPort(7),
resetConnUnitPortCounters(8)
}
```

Access

read-write

Status

mandatory

Return Value

Table 23 lists the connUnitPortControl read return values.

Table 23. ConnUnitPortControl read return values

Port Value	Return Value
Online	online (7)
Offline	offline (6)
Diagnostics	offline (6)
other	unknown (1)

Refer to Table 24 for connUnitPortControl write command values.

Table 24. ConnUnitPortControl write command values

Control Value	Command Sent
Online (7)	online
Offline (6)	offline
ResetCounters (8)	clear counters
other	error returned

connUnitPortName (1.3.6.1.3.94.1.10.1.17)

A user-defined name for this port. This means that up to DisplayString characters may be supported. If less than, then the name will be truncated in the connunit.

Syntax INTEGER

Access read-write

Status mandatory

Return Value Symbolic port name. A 1G or 2G only capable port, would return port followed by the port number. 10G ports would return 10G followed by the port number. For example, a 1G/2G port#2 would return 'Port2' and a 10G port#18 would return '10G-18' by default.

connUnitPortPhysicalNumber (1.3.6.1.3.94.1.10.1.18)

This is the internal port number this port is known by. In many implementations, this should be the same as connUnitPortIndex. Some implementations may have an internal port representation not compatible with the rules for table indexes. In that case, provide the internal representation of this port in this object. This value may also be used in the connUnitLinkPortNumberX or connUnitLinkPortNumberY objects of the connUnitLinkTable.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Physical port number.

connUnitPortStatObject (1.3.6.1.3.94.1.10.1.19)

This contains the OID of the first object of the table that contains the statistics for this particular port. If this has a value of zero, then there are no statistics available for this port. The port type information will help identify the statistics objects that will be found in the table.

Syntax OBJECT IDENTIFIER

Access read-only

Status deprecated

Return Value Port object ID (1.2.6.1.3.94.4.5.1.1).

connUnitPortProtocolCap (1.3.6.1.3.94.1.10.1.20)

Bit mask that specifies the driver level protocol capability of this port. If this is not applicable, returns all bits set to zero.

The bits have the following definitions:

unknown - 0

Loop - 1

Fabric - 2

SCSI - 4

TCP/IP - 8

VI - 16

FICON - 32

Syntax OCTET STRING (SIZE (2))

Access read-only

Status mandatory

Return Value 0x03 (Loop, Fabric).

connUnitPortProtocolOp (1.3.6.1.3.94.1.10.1.21)

Bit mask that specifies the driver level protocol(s) that are currently operational. If not applicable, return all bits set to zero. This object has the same definition as connUnitPortProtocolCap.

Syntax OCTET STRING (SIZE (2))

Access read-only

Status unsupported

Return Value Error status `NoSuchName`

connUnitPortNodeWwn (1.3.6.1.3.94.1.10.1.22)

The Node World Wide Name of the port if applicable, otherwise all zeros. This should have the same value for a group of related ports. The container is defined as the largest physical entity. For example, all ports on HBAs on a host will have the same Node WWN. All ports on the same storage subsystem will have the same Node WWN." ::= { connUnitPortEntry 22 }.

Syntax FcNameId

Access read-only

Status mandatory

Return Value World Wide Node Name of the switch. For example: 10 00 00 C0 DD 00 71 C9.

connUnitPortHWState (1.3.6.1.3.94.1.10.1.23)

The hardware detected state of the port.

Syntax

INTEGER {
unknown(1),
failed(2), - port failed diagnostics
bypassed(3), - FCAL bypass, loop only
active(4), - connected to a device
loopback(5), - Port in external loopback
txfault(6), - Transmitter fault
noMedia(7), - media not installed linkDown
(8) - waiting for activity (rx sync)
}

Access

read-only

Status

mandatory

Return Value

Table 25 lists the connUnitPortHWState port state return values.

Table 25. ConnUnitPortHWState port state return values

Port state	Return value
If DiagStatus = Failed	Failed (2)
If SFP = Not Installed	NoMedia (7)
If SyncStatus = SyncAcquired	Active (4)
If SyncStatus = SyncLost	LinkDown (8)
Other	Unknown (1)

Event table

The objects described in this section are in a table format indexed by World Wide Name and Index. The maximum index is determined based on the number of events in the table. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitEventTable.connUnitEventEntry.connUnitEventUnitId
.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

connUnitEventUnitId (1.3.6.1.3.94.1.11.1.1)

The connUnitId of the connectivity unit that contains this event table.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitEventIndex (1.3.6.1.3.94.1.11.1.2)

Each connectivity unit has its own event buffer. As it wraps, it may write over previous events. This object is an index into the buffer. It is recommended that this table be read using getNext's to retrieve the initial table. The management application should read the event table at periodic intervals and then determine if any new entries were added by comparing the last known index value with the current highest index value. The management application should then update its copy of the event table. If the read interval is too long, it is possible that there may be events that may not be contained in the agent's internal event buffer. For example, an agent may read events 50-75. At the next read interval, connUnitEventCurrID is 189. If the management application tries to read event index 76, and the agent's internal buffer is 100 entries max, event index 76 will no longer be available.

The index value is an incrementing integer starting from one every time there is a table reset. On table reset, all contents are emptied and all indexes are set to zero. When an event is added to the table, the event is assigned the next higher integer value than the last item entered into the table. If the index value reaches its maximum value, the next item entered will cause the index value to roll over and start at one again.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Table index

connUnitEventId (1.3.6.1.3.94.1.11.1.3)

The internal event ID. Incremented for each event, ranging between 1 and connUnitMaxEvents. Not used as table index to simplify the agent implementation. When this reaches the end of the range specified by connUnitMaxEvents, the ID will roll over to start at one. This value will be set back to one at reset. The relationship of this value to the index is that internal event ID may represent a smaller number than a 32 bit integer (for example, maximum 100 entries) and would only have a value range up to connUnitMaxEvents.

Syntax INTEGER

Access read-only

Status deprecated

Return Value Unsupported. Always returns error status `NoSuchName`.

connUnitREventTime (1.3.6.1.3.94.1.11.1.4)

The real time when the event occurred. It has the following format.

DDMMYYYY HHMMSS

DD=day number

MM=month number

YYYY=year number

HH=hour number

MM=minute number

SS=seconds number

If not applicable, return either a NULL string or “00000000 000000”.

Syntax DisplayString (SIZE (0..15))

Access read-only

Status mandatory

Return Value Timestamp of the event

connUnitSEventTime (1.3.6.1.3.94.1.11.1.5)

This is the sysUpTime timestamp when the event occurred.

Syntax connUnitSEventTime

Access read-only

Status mandatory

Return Value Error status `NoSuchName`

connUnitEventSeverity (1.3.6.1.3.94.1.11.1.6)

The event severity level.

Syntax FcEventSeverity

Access read-only

Status mandatory

Return Value Error status `NoSuchName`

connUnitEventType (1.3.6.1.3.94.1.11.1.7)

The type of this event.

Syntax

INTEGER {
unknown(1),
other(2),
status(3),
configuration(4),
topology(5)
}

Access

read-only

Status

mandatory

Return Value

3 (Status)

connUnitEventObject (1.3.6.1.3.94.1.11.1.8)

This is used with the connUnitEventType to identify the object that is referenced by the event. Examples include connUnitPortStatus.connUnitId.connUnitPortIndex and connUnitStatus.connUnitId.

Syntax OBJECT IDENTIFIER

Access read-only

Status mandatory

Return Value Error status `NoSuchName`

connUnitEventDescr (1.3.6.1.3.94.1.11.1.9)

The description of the event.

Syntax DisplayString

Access read-only

Status mandatory

Return Value Event description in the form:
`[Id][timestamp][severity][module][Description]`

Link table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index is an index into the link table for the switch. There may be as many link entries as there are ports. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitLinkTable.connUnitLinkEntry.connUnitLinkUnitId.16
.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

If the agent is able to discover links that do not directly attach to members of its agency and its discovery algorithm gives some assurance the links are recently valid, it may include these links. Link information entered by administrative action may be included even if not validated directly if the link has at least one endpoint in this agency, but should not be included otherwise.

A connectivity unit should fill the table in as best it can. One of the methods to fill this in would be to use the RNID ELS (ANSI document 99-422v0). This allows one to query a port for the information needed for the link table.

This table is accessed either directly if the management software has an index value or via GetNexts. The value of the indexes are not required to be contiguous. Each entry created in this table will be assigned an index. This relationship is kept persistent until the entry is removed from the table or the system is reset. The total number of entries are defined by the size of the table

connUnitLinkUnitId (1.3.6.1.3.94.1.12.1.1)

The connUnitId of the connectivity unit that contains this link table.

Syntax connUnitLinkUnitId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitLinkIndex (1.3.6.1.3.94.1.12.1.2)

This index is used to create a unique value for each entry in the link table with the same connUnitLinkUnitId. The value can only be reused if it is not currently in use and the value is the next candidate to be used. This value wraps at the highest value represented by the size of INTEGER. This value is reset to zero when the system is reset, and the first value to be used is one.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Table index.

connUnitLinkNodeIdX (1.3.6.1.3.94.1.12.1.3)

The Node WWN of the unit at one end of the link. If the Node WWN is unknown and the Node is a connUnit in the responding agent, then the value of this object must be equal to its connUnitID.

Syntax OCTET STRING (SIZE(16))

Access read-only

Status mandatory

Return Value World Wide Name of the local switch for each entry in the link table. For example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitLinkPortNumberX (1.3.6.1.3.94.1.12.1.4)

The port number on the unit specified by connUnitLinkNodeIdX if known, otherwise -1. If the value is non-negative, then it will be equal to connUnitPortPhysicalNumber.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Local port number for each entry in the link table

connUnitLinkPortWwnX (1.3.6.1.3.94.1.12.1.5)

The port WWN of the unit specified by connUnitLinkNodeIdX if known, otherwise 16 octets of binary 0" ::= { connUnitLinkEntry 5 }.

Syntax connUnitLinkPortWwnX

Access read-only

Status mandatory

Return Value Local World Wide port number for each entry in the link table

connUnitLinkNodeIdY (1.3.6.1.3.94.1.12.1.6)

The Node WWN of the unit at the other end of the link. If the Node WWN is unknown and the Node is a connUnit in the responding SNMP agency, then the value of this object must be equal to its connUnitID.

Syntax OCTET STRING (SIZE(16))

Access read-only

Status mandatory

Return Value Remote World Wide Node number for each entry in the link table

connUnitLinkPortNumberY (1.3.6.1.3.94.1.12.1.7)

The port number on the unit specified by connUnitLinkNodeIdY if known, otherwise -1. If the value is non-negative, then it will be equal to connUnitPortPhysicalNumber.

Syntax OCTET STRING (SIZE(16))

Access read-only

Status mandatory

Return Value Remote port number for inter-switch link, if known. Otherwise, -1 (0xFFFFFFFF)

connUnitLinkPortWwnY (1.3.6.1.3.94.1.12.1.8)

The port WWN on the unit specified by connUnitLinkNodeIdY if known, otherwise 16 octets of binary 0" ::= { connUnitLinkEntry 8 }.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value Remote Port World Wide Name for each entry in the link table, if known

connUnitLinkAgentAddressY (1.3.6.1.3.94.1.12.1.9)

The address of an FCMGMT MIB agent for the Node identified by connUnitLinkNodeIdY, if known. Otherwise 16 octets of binary 0" ::= {connUnitLinkEntry 9}.

Syntax OCTET STRING (SIZE(16))

Access read-only

Status mandatory

Return Value Remote IP address of the remote switch, if known. Otherwise, returns sixteen zeroes.

connUnitLinkAgentAddressTypeY (1.3.6.1.3.94.1.12.1.10)

If connUnitLinkAgentAddressY is nonzero, it is a protocol address. ConnUnitLinkAgentAddressTypeY is the “address family number” assigned by IANA to identify the address format.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 1 (Ipv4)

connUnitLinkAgentPortY (1.3.6.1.3.94.1.12.1.11)

The IP port number for the agent. This is provided in case the agent is at a non-standard SNMP port.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 0

connUnitLinkUnitTypeY (1.3.6.1.3.94.1.12.1.12)

Type of the Fibre Channel connectivity unit as defined in connUnitType.

Syntax FcUnitType

Access read-only

Status mandatory

Return Value Remote device in the link table, for example, switch (4)

connUnitLinkConnIdY (1.3.6.1.3.94.1.12.1.13

This is the Fibre Channel ID of this port. If the connectivity unit is a switch, this is expected to be a Big Endian value of 24 bits. If this is loop, then it is the ALPA that is connected. If this is an E_Port, then it will only contain the domain ID. If not any of those, unknown or cascaded loop, returns all bits set to 1.

Syntax OCTET STRING (SIZE(3))

Access read-only

Status mandatory

Return Value Remote Fibre Channel address of each entry in the link table

connUnitLinkCurrIndex (1.3.6.1.3.94.1.12.1.14)

The last used link index.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Last used link table index number

Zone Table

The objects described in this section are in a table format indexed Zone number and Index. The zones are numbered 1 to connUnitZoneSetNumZones, the index represents the members within the zones.

An example of how to access one of these objects:

fcmgmt.connSet.connUnitZoneTable.connUnitZoneEntry.connUnitZoneIndex.1.1

connUnitZoneIndex (1.3.6.1.3.94.1.13.1.1)

Unique table index for each zone. Valid values are between 1 and connUnitZoneSetNumZones.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Index number for each zone within the active zone set

connUnitZoneMemberIndex (1.3.6.1.3.94.1.13.1.2)

Unique table index for each zone member. Valid values are between 1 and connUnitZoneNumMembers.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Index number for each member within a zone

connUnitZoneSetName (1.3.6.1.3.94.1.13.1.3)

Name of the active zone set to which the zone and zone member belong.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Zone set name

connUnitZoneSetNumZones (1.3.6.1.3.94.1.13.1.4)

The number of zones in the active zone set.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of zones within the active zone set

connUnitZoneName (1.3.6.1.3.94.1.13.1.5)

Name of the zone.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Zone name

connUnitZoneCapabilities (1.3.6.1.3.94.1.13.1.6)

1-byte bit mask that specifies the zoning capabilities supported by the fabric.

Bit 7 - Soft zones supported.

Bit 6 - Hard zones supported.

Bits 5-0 - Reserved.

Syntax OCTET STRING (SIZE(1))

Access read-only

Status mandatory

Return Value 0xC0

connUnitZoneEnforcementState (1.3.6.1.3.94.1.13.1.7)

1-byte bit mask that specifies the current enforcement of the Zone Set.

Bit 7 - Soft zone set enforced.

Bit 6 - Hard zone set enforced.

Bits 5-0 - Reserved.

Syntax OCTET STRING (SIZE(1))

Access read-only

Status mandatory

Return Value Zone type: soft (0x80), hard (0x40)

connUnitZoneAttributeBlock (1.3.6.1.3.94.1.13.1.8)

A variable length structure that contains extended zone attributes defined in the FC-GS-4 enhanced zone server. See FC-GS-4 draft standard for details and format of the structure. Support of this object is optional.

Syntax OCTET STRING (SIZE(80))

Access read-only

Status mandatory

Return Value Not supported. Always returns SNMP error `NoSuchName`.

connUnitZoneNumMembers (1.3.6.1.3.94.1.13.1.9)

Number of zone members in the zone: connUnitZoneName.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Total number of members in a zone.

connUnitZoneMemberIdType (1.3.6.1.3.94.1.13.1.10)

Type of zone member ID:

1- Port WWN
2- Domain & Port ID
3- FC Address
4- Node WWN
5- Alias Name
6-'FF'h - Vendor specified.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Member ID type, which is mapped as follows:

- WWN—0x01 // Port WWN
- Domain/Port—0x02 // Domain and port ID
- FCaddress—0x03 // FC address
- [other]—0xff // Vendor specific

connUnitZoneMemberID (1.3.6.1.3.94.1.13.1.11)

ID of the zone member based on connUnitZoneMemberIdType.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value Zone member name as a 16 8-bit octets, which is mapped as follows:

- WWN member—WWN (8 bytes) followed by 8 bytes of zeros.
- FC address—FC address (3 bytes) followed by 13 bytes of zeros.
- Domain/Port—Domain/port address (2 bytes) followed by 14 bytes of zeros.

Zoning alias table

The objects described in this section are in a table format indexed by Alias Number and Index. The aliases are numbered 1 to connUnitZoningAliasNumAliases, the index represents the members within the alias. An example of how to access one of these objects:

```
"fcmgmt.connSet.connUnitZoneTable.connUnitZoneEntry.connUnitZoningAliasIndex.1.1"
```

connUnitZoningAliasIndex (1.3.6.1.3.94.1.14.1.1)

Unique table index for each alias. Valid values are between 1 and connUnitZoningAliasNumAliases.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Alias index

connUnitZoningAliasMemberIndex (1.3.6.1.3.94.1.14.1.2)

Unique table index for each alias member. Valid values are between 1 and connUnitZoningAliasNumMembers.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Alias member index

connUnitZoningAliasNumAliases (1.3.6.1.3.94.1.14.1.3)

The number of aliases defined in this table.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of aliases defined

connUnitZoningAliasName (1.3.6.1.3.94.1.14.1.4)

The alias name.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Alias name

connUnitZoningAliasNumMembers (1.3.6.1.3.94.1.14.1.5)

Number of members in the alias: connUnitZoningAliasName.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of members in a defined alias zone

connUnitZoningAliasMemberIdType (1.3.6.1.3.94.1.14.1.6)

Type of alias member ID:

1—Port WWN

2—Domain and port ID

3—FC address

Others—reserved.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Alias member ID type, which is mapped as follows:

- WWN—0x01 // Port WWN
- DomainPort—0x02 // Domain and port ID
- FC Address—0x03 // FC address
- [other]—0xff // Vendor specific

connUnitZoningAliasMemberID (1.3.6.1.3.94.1.14.1.7)

ID of the alias member based on connUnitZoningAliasMemberIdType.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value Alias zone member name as 16 8-bit octets, which is mapped as follows:

- WWN member—WWN (8 bytes) followed by 8 bytes of zeros.
- FC address—FC address (3 bytes) followed by 13 bytes of zeros.
- Domain/Port—Domain/Port address (2 bytes) followed by 14 bytes of zeros.

Port statistics table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index represents the port number to interrogate. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.statSet.connUnitPortStatTable.connUnitPortStatEntry.connUnitPortS
tatUnitId.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

There is one and only one statistics table for each individual port. For all objects in statistics table, if the object is not supported by the conn unit then the high order bit is set to 1 with all other bits set to zero. The high order bit is reserved to indicate if the object if supported or not. All objects start at a value of zero at hardware initialization and continue incrementing till end of 63 bits and then wrap to zero.

connUnitPortStatUnitId (1.3.6.1.3.94.4.5.1.1)

A unique value among all entries in this table having the same connUnitPortStatUnitId, between 1 and connUnitNumPort [connUnitPortStatUnitId].

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitPortStatIndex (1.3.6.1.3.94.4.5.1.2)

A unique value among all entries in this table, between 0 and connUnitNumPort[connUnitPortUnitId].

Syntax INTEGER (0..2147483647)

Access read-only

Status mandatory

Return Value Port table index

connUnitPortStatCountError (1.3.6.1.3.94.4.5.1.3)

A count of the errors that have occurred on this port.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of port errors expressed as a hexidecimal value

connUnitPortStatCountTxObjects (1.3.6.1.3.94.4.5.1.4)

The number of frames/packets/IOs/etc transmitted by this port. A Fibre Channel frame starts with SOF and ends with EOF. Fibre Channel loop devices should not count frames passed through. This value represents the sum total for all other Tx objects.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of bytes transmitted by a port expressed as a hexidecimal value

connUnitPortStatCountRxObjects (1.3.6.1.3.94.4.5.1.5)

The number of frames/packets/IOs/etc received by this port. A Fibre Channel frame starts with SOF and ends with EOF. Fibre Channel loop devices should not count frames passed through. This value represents the sum total for all other Rx objects.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of bytes received by a port expressed as a hexidecimal value

connUnitPortStatCountTxElements (1.3.6.1.3.94.4.5.1.6)

The number of octets or bytes that have been transmitted by this port. One second periodic polling of the port. This value is saved and compared with the next polled value to compute net throughput. For Fibre Channel, ordered sets are not included in the count.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of bytes transmitted by a port expressed as a hexidecimal value

connUnitPortStatCountRxElements (1.3.6.1.3.94.4.5.1.7)

The number of octets or bytes that have been received by this port. One second periodic polling of the port. This value is saved and compared with the next polled value to compute net throughput. For Fibre Channel, ordered sets are not included in the count.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of bytes received by a port expressed as a hexidecimal value

connUnitPortStatCountBBCreditZero (1.3.6.1.3.94.4.5.1.8)

Count of transitions in/out of BBcredit zero state. The other side is not providing any credit. This is a Fibre Channel statistic only.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountInputBuffersFull (1.3.6.1.3.94.4.5.1.9)

Count of occurrences when all input buffers of a port were full and outbound buffer-to-buffer credit transitioned to zero. There is no credit to provide to other side. This is a Fibre Channel statistic only.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountFBSYFrames (1.3.6.1.3.94.4.5.1.10)

Count of times that FBSY was returned to this port as a result of a frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. Port can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic. This is the sum of all classes. If you cannot keep the by-class counters, then keep the sum counters.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of FBusy on a port expressed as a hexidecimal value

connUnitPortStatCountPBSYFrames (1.3.6.1.3.94.4.5.1.11)

Count of times that PBSY was returned to this port as a result of a frame that could not be delivered to the other end of the link. This occurs if the destination port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic. This is the sum of all classes. If you cannot keep the by-class counters, then keep the sum counters.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit set to 1 with remaining bits set to zero.

connUnitPortStatCountFRJTFrames (1.3.6.1.3.94.4.5.1.12)

Count of times that FRJT was returned to this port as a result of a frame that was rejected by the fabric. This is the total for all classes and is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of Frame Rejects on a port expressed as hexidecimal value

connUnitPortStatCountPRJTFrames (1.3.6.1.3.94.4.5.1.13)

Count of times that FRJT was returned to this port as a result of a frame that was rejected at the destination N_Port. This is the total for all classes and is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1RxFrames (1.3.6.1.3.94.4.5.1.14)

Count of Class 1 frames received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1TxFrames (1.3.6.1.3.94.4.5.1.15)

Count of Class 1 frames transmitted from this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1FBSYFrames (1.3.6.1.3.94.4.5.1.16

Count of times that FBSY was returned to this port as a result of a Class 1 frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. FBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1PBSYFrames (1.3.6.1.3.94.4.5.1.17)

Count of times that PBSY was returned to this port as a result of a Class 1 frame that could not be delivered to the other end of the link. This occurs if the destination N_Port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1FRJTFrames (1.3.6.1.3.94.4.5.1.18

Count of times that FRJT was returned to this port as a result of a Class 1 frame that was rejected by the fabric. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass1PRJTFrames (1.3.6.1.3.94.4.5.1.19)

Count of times that FRJT was returned to this port as a result of a Class 1 frame that was rejected at the destination N_Port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass2RxFrames (1.3.6.1.3.94.4.5.1.20)

Count of Class 2 frames received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of Class 2 frames received by a port

connUnitPortStatCountClass2TxFrames (1.3.6.1.3.94.4.5.1.21)

Count of Class 2 frames transmitted from this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of Class 2 frames transmitted by a port

connUnitPortStatCountClass2FBSYFrames (1.3.6.1.3.94.4.5.1.22)

Count of times that FBSY was returned to this port as a result of a Class 2 frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. FBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass2PBSYFrames (1.3.6.1.3.94.4.5.1.23)

Count of times that PBSY was returned to this port as a result of a Class 2 frame that could not be delivered to the other end of the link. This occurs if the destination N_Port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass2FRJTFrames (1.3.6.1.3.94.4.5.1.24)

Count of times that FRJT was returned to this port as a result of a Class 2 frame that was rejected by the fabric. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass2PRJTFrames (1.3.6.1.3.94.4.5.1.25)

Count of times that FRJT was returned to this port as a result of a Class 2 frame that was rejected at the destination N_Port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountClass3RxFrames (1.3.6.1.3.94.4.5.1.26)

Count of Class 3 frames received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value The total number of Class 3 frames received by a port.

connUnitPortStatCountClass3TxFrames (1.3.6.1.3.94.4.5.1.27)

Count of Class 3 frames transmitted from this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value The total number of Class 3 frames transmitted by a port.

connUnitPortStatCountClass3Discards (1.3.6.1.3.94.4.5.1.28)

Count of Class 3 frames that were discarded upon reception at this port. There is no FBSY or FRJT generated for Class 3 frames. They are simply discarded if they cannot be delivered. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value The total number of Class3Toss frames for a port.

connUnitPortStatCountRxMulticastObjects (1.3.6.1.3.94.4.5.1.29)

Count of Multicast frames or packets received at this port.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountTxMulticastObjects (1.3.6.1.3.94.4.5.1.30)

Count of Multicast frames or packets transmitted from this port.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountRxBroadcastObjects (1.3.6.1.3.94.4.5.1.31)

Count of Broadcast frames or packets received at this port.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountTxBroadcastObjects (1.3.6.1.3.94.4.5.1.32)

Count of Broadcast frames or packets transmitted from this port. On a Fibre Channel loop, count only OPNr frames generated.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountRxLinkResets (1.3.6.1.3.94.4.5.1.33)

Count of link resets. This is the number of LRs received. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of RxLinkResets received by a port

connUnitPortStatCountTxLinkResets (1.3.6.1.3.94.4.5.1.34)

Count of link resets. The number of LRs transmitted. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of TxLinkResets transmitted by a port

connUnitPortStatCountNumberLinkResets (1.3.6.1.3.94.4.5.1.35)

Count of link resets and LIPs detected at this port. The number of times the reset link protocol is initiated. These are the count of the logical resets, and a count of the number of primitives. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of TotalLinkResets for a port

connUnitPortStatCountRxOfflineSequences (1.3.6.1.3.94.4.5.1.36)

Count of offline primitive OLSs received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of RxOfflineSeqs received by a port

connUnitPortStatCountTxOfflineSequences (1.3.6.1.3.94.4.5.1.37)

Count of offline primitive OLSs transmitted by this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of TxOfflineSeqs transmitted by a port

connUnitPortStatCountNumberOfflineSequences (1.3.6.1.3.94.4.5.1.38)

Count of offline primitive sequences received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of TotalOfflineSeqs received by a port

connUnitPortStatCountLinkFailures (1.3.6.1.3.94.4.5.1.39)

Count of link failures. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of LinkFailures for a port

connUnitPortStatCountInvalidCRC (1.3.6.1.3.94.4.5.1.40)

Count of frames received with invalid CRC. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). Loop ports should not count CRC errors passing through when monitoring. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of InvalidCRCs received by a port

connUnitPortStatCountInvalidTxWords (1.3.6.1.3.94.4.5.1.41)

Count of invalid transmission words received at this port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of DecodeErrors for a port

connUnitPortStatCountPrimitiveSequenceProtocolErrors (1.3.6.1.3.94.4.5.1.42)

Count of primitive sequence protocol errors detected at this port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of PrimSeqErrors for a port

connUnitPortStatCountLossofSignal (1.3.6.1.3.94.4.5.1.43)

Count of instances of signal loss detected by the port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountLossofSynchronization (1.3.6.1.3.94.4.5.1.44)

Count of instances of synchronization loss detected by the port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of LossOfSyncs detected by this port

connUnitPortStatCountInvalidOrderedSets (1.3.6.1.3.94.4.5.1.45)

Count of invalid ordered sets received by the port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountFramesTooLong (1.3.6.1.3.94.4.5.1.46)

Count of frames received at this port where the frame length was greater than what was agreed to in FLOGI/PLOGI. This could be caused by losing the end of frame delimiter. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountFramesTruncated (1.3.6.1.3.94.4.5.1.47)

Count of frames received at this port where the frame length was less than the minimum indicated by the frame header (normally 24 bytes). It could be more if the DFCTL field indicates an optional header should have been present. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountAddressErrors (1.3.6.1.3.94.4.5.1.48)

Count of frames received with unknown addressing.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Total number of InvalidDestAddr frames received by a port

connUnitPortStatCountDelimiterErrors (1.3.6.1.3.94.4.5.1.49)

Count of invalid frame delimiters received at this port. An example is a frame with a Class 2 start and a Class 3 at the end. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

connUnitPortStatCountEncodingDisparityErrors (1.3.6.1.3.94.4.5.1.50)

Count of disparity errors received at this port. This is a Fibre Channel-only statistic.

Syntax OCTET STRING (SIZE (8))

Access read-only

Status mandatory

Return Value Unsupported. Always returns high order bit to 1 with all other bits set to zero.

Simple Name Server Table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index represents the table index. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connUnitServiceSet.connUnitServiceTables.connUnitSnsTable.connUni
tSnsEntry.connUnitSnsId.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

The Fibre Channel Simple Name Server table contains an entry for each device presently known to this connUnit. There will not be any version on this since FC-GS3 does not define a version today.

This table is accessed either directly if the management software has an index value or using GetNexts. The value of the indexes are not required to be contiguous. Each entry created in this table will be assigned an index. This relationship is kept persistent until the entry is removed from the table or the system is reset. The total number of entries are defined by the size of the table.

connUnitSnsMaxEntry (1.3.6.1.3.94.5.1.1)

The current number of entries in the table.

Syntax INTEGER

MaxAccess read-only

Status mandatory

Return Value Number of entries registered in the Simple Name Server for all switches

connUnitSnsId (1.3.6.1.3.94.5.2.1.1.1)

The connUnitId of the connectivity unit that contains this Name Server table.

Syntax OCTET STRING (SIZE (16))

Access read-only

Status mandatory

Return Value World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00.

connUnitSnsPortIndex (1.3.6.1.3.94.5.2.1.1.2)

The physical port number of this SNS table entry. Each physical port has an SNS table with 1-n entries indexed by ConnUnitSnsPortIdentifier (port address).

Syntax INTEGER

Access read-only

Status mandatory

Return Value Name server table index

connUnitSnsPortIdentifier (1.3.6.1.3.94.5.2.1.1.3)

The port identifier for this entry in the SNS table.

Syntax FcAddressId

Access read-only

Status mandatory

Return Value 24-bit Fibre Channel address for each entry in the name server table based on Domain, Area, and ALPA

connUnitSnsPortName (1.3.6.1.3.94.5.2.1.1.4)

The Port World Wide Name for this entry in the SNS table.

Syntax FcNameId

Access read-only

Status mandatory

Return Value Port World Wide Name of the device in the name server table

connUnitSnsNodeName (1.3.6.1.3.94.5.2.1.1.5)

The Node name for this entry in the SNS table.

Syntax FcNameId

Access read-only

Status mandatory

Return Value Node World Wide Name of the device in the name server table

connUnitSnsClassOfSvc (1.3.6.1.3.94.5.2.1.1.6)

The classes of service offered by this entry in the SNS table.

Syntax OCTET STRING (SIZE (1))

Access read-only

Status mandatory

Return Value First registered class of service for an entry in the name server table. This is a bit mask where each bit that represents the class of service is set to a value of one if the class is supported. Class 1 is bit zero.

connUnitSnsNodeIPAddress (1.3.6.1.3.94.5.2.1.1.7)

The IPv6 formatted address of the Node for this entry in the SNS table.

Syntax OCTET STRING (SIZE (16))

Access read-only

Status mandatory

Return Value Switch IP address in IPv6 format

connUnitSnsProcAssoc (1.3.6.1.3.94.5.2.1.1.8)

The process associator for this entry in the SNS table.

Syntax OCTET STRING (SIZE (16))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitSnsFC4Type (1.3.6.1.3.94.5.2.1.1.9)

The FC-4 types supported by this entry in the SNS table.

Syntax OCTET STRING (SIZE (32))

Access read-only

Status mandatory

Return Value FC-4 Types registered for the device in the name server table. This is a 32 byte field with each bit uniquely identifying the FC-4 Type registered as defined in FC-GS-3 specification. Example: SCSI FCP (bit 8) = 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00.

connUnitSnsPortType (1.3.6.1.3.94.5.2.1.1.10)

The port type of this entry in the SNS table.

Syntax OCTET STRING (SIZE (1))

Access read-only

Status mandatory

Return Value PortType for the entry in the name server table. Table 26 lists the connUnitPortType port type return values.

Table 26. ConnUnitPortType state return values

Port type	Return value (hexidecimal)
N	1
NL	2
F/NL	3
NX	7F
F	8
FL	82
E	84
B	85

connUnitSnsPortIPAddress (1.3.6.1.3.94.5.2.1.1.11)

The IPv6 formatted address of this entry in the SNS table.

Syntax OCTET STRING (SIZE (16))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitSnsFabricPortName (1.3.6.1.3.94.5.2.1.1.12)

The fabric port name of this entry in the SNS table.

Syntax FcNameId

Access read-only

Status mandatory

Return Value Switch port Port World Wide Name for the device in the name server table

connUnitSnsHardAddress (1.3.6.1.3.94.5.2.1.1.13)

The hard ALPA of this entry in the SNS table.

Syntax FcAddressId

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitSnsSymbolicPortName (1.3.6.1.3.94.5.2.1.1.14)

The symbolic port name of this entry in the SNS table.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Symbolic port name registered by the device in the name server table. If not registered, returns (NULL).

connUnitSnsSymbolicNodeName (1.3.6.1.3.94.5.2.1.1.15)

The symbolic Node name of this entry in the SNS table.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Symbolic node name registered by the device in the name server table. If not registered, returns (NULL).

Platform Table

The Platform Table is a simple, read-only view of platform registration entries. Platform registry is a service hosted by the connectivity unit, in a very similar manner as the SNS table. The platform table is contained by the connectivity unit. A platform can register its attributes and platform nodes with the registry service.

The platform table is a flat, double-indexed MIB table. To keep the table simple, only one platform management URL is exposed. If a platform registers more than one management URL, the first one is reported in this table. This table is based on the fabric configuration server defined in the FC-GS-3 standard and enhanced platform attributes proposed for FC-GS-4. Note that the information contained in this table may only contain the platforms that this connUnit can see or it may contain a fabric wide view of the platforms.

connUnitPlatformMaxEntry (1.3.6.1.3.94.5.1.2)

The maximum number of entries in the platform table.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformIndex (1.3.6.1.3.94.5.2.2.1.1)

Unique table index for each platform. Valid values are between 1 and connUnitPlatformsMaxEntry.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformNodeIndex (1.3.6.1.3.94.5.2.2.1.2)

Unique table index for each platform node. Valid values are between 1 and connUnitPlatformsNumNodes.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformUnitID (1.3.6.1.3.94.5.2.2.1.3)

The connUnitId of the connectivity unit that contains this Platform table.

Syntax FcGlobalId

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformName (1.3.6.1.3.94.5.2.2.1.4)

The platform name. May be either a readable string or a unique ID format as specified in the FC-GS-4 draft standard.

Syntax OCTET STRING (SIZE(79))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformType (1.3.6.1.3.94.5.2.2.1.6)

The platform type.

Syntax FcUnitType

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformLabel (1.3.6.1.3.94.5.2.2.1.7)

An administratively assigned symbolic name for the platform. The Platform Label shall only contain print-able ASCII characters.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformDescription (1.3.6.1.3.94.5.2.2.1.8)

A textual description of the platform. This value should include the full name and version identification of the platform's hardware type and software operating system. The Platform Description shall only contain printable ASCII characters.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`

connUnitPlatformLocation (1.3.6.1.3.94.5.2.2.1.9)

The physical location of the platform (e.g., telephone closet, 3rd floor). The Platform Location shall only contain printable ASCII characters.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`.

connUnitPlatformManagementUrl (1.3.6.1.3.94.5.2.2.1.10)

Primary management URL for the platform. If the platform registers more than one URL, then this URL is equal to the first in the list.

Syntax DisplayString (SIZE (0..79))

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`.

connUnitPlatformNumNodes (1.3.6.1.3.94.5.2.2.1.11)

Number of nodes contained in the platform.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Unsupported. Always returns error status `NoSuchName`.

connUnitPlatformNodeName (1.3.6.1.3.94.5.2.2.1.12)

The name (WWN - world wide name) of the node contained by the platform.

Syntax FcGlobalId

Access read-only

Status read-only

Return Value Unsupported. Always returns error status `NoSuchName`.

Trap Table

Traps are asynchronous messages sent from the agent (residing on the switch) to the manager (residing on the workstation) to identify significant events.

There can be up to 5 trap addresses within the trap table. All trap information is stored within the switch and is accessible to Telnet and the SNMP agent, and is persistent between boots. An example of how to access one of these objects given an IP address of 10.32.165.4 is:

```
"snmpget localhost public
fcmgmt.trapReg.trapRegTable.trapRegEntry.trapRegFilter.10.32.165.4.162".
```

A trap event is reported when the incoming error has a severity level less than or equal to the configured severity level. The trap event types and trap severity levels are listed in Table 27.

Table 27. Trap severity levels

Event Type	Severity Level
Unknown	1
Emergency	2
Alert	3
Critical	4
Error	5
Warning	6
Notify	7
Info	8
Debug	9
Mark	10

trapMaxClients (1.3.6.1.3.94.2.1)

The maximum number of SNMP trap recipients supported by the connectivity unit.

Syntax INTEGER

Access read-only

Status mandatory

Return Value 5

trapClientCount (1.3.6.1.3.94.2.2)

The current number of rows in the trap table.

Syntax INTEGER

Access read-only

Status mandatory

Return Value Number of configured trap clients: 1–5

trapRegIpAddress (1.3.6.1.3.94.2.3.1.1)

The IP address of a client registered for traps.

Syntax IpAddress

Access read-only

Status mandatory

Return Value IP addresses (as defined in the trap table) to which to send traps when they occur

trapRegPort (1.3.6.1.3.94.2.3.1.2)

The UDP port to send traps to for this host. Normally this would be the standard trap port (162). This object is an index and must be specified to create a row in this table.

Syntax INTEGER (1..2147483647)

Access read-only

Status mandatory

Return Value Configured port number to which to send traps when they occur. The port number can be configured in the switch SNMP setup parameters. Default is 162.

trapRegFilter (1.3.6.1.3.94.2.3.1.3)

This value defines the trap severity filter for this trap host. The connUnit will send traps to this host that have a severity level less than or equal to this value. The default value of this object is “warning”.

Syntax FcEventSeverity

Access read-write

Status mandatory

Return Value Trap severity level as listed in Table 27

trapRegRowState (1.3.6.1.3.94.2.3.1.4)

Specifies the state of the row.

- rowDestroy
  - READ: Can never happen.
  - WRITE: Remove this row from the table.
- rowInactive
  - READ: Indicates that this row does exist, but that traps are not enabled to be sent to the target.
  - WRITE: If the row does not exist, and the agent allows writes to the trap table, then a new row is created. The values of the optional columns will be set to default values. Traps are not enabled to be sent to the target. If the row already existed, then traps are disabled from being sent to the target.
- rowActive
  - READ: Indicates that this row exists, and that traps are enabled to be sent to the target.
  - WRITE: If the row does not exist, and the agent allows writes to the trap table, then a new row is created. The values of the optional columns will be set to default values. Traps are enabled to be sent to the target. If the row already exists, then traps are enabled to be sent to the target.

  A value of “rowActive” or “rowInactive” must be specified to create a row in the table.

Syntax

INTEGER {
rowDestroy(1), - Remove row from table.
rowInactive(2), - Row exists, but traps disabled
rowActive(3) - Row exists and is enabled for sending traps
}

Access

read-write

Status

mandatory

Return Value

rowActive (3), if valid entry in trap table.

Related traps

The following traps contain the trap information being sent from the agent to the manager.

connUnitStatusChange (1.3.6.1.3.94.0.1)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever a Switch.OperChange or Switch.StateChange event occurs.

Variables: { connUnitStatus, connUnitState }

connUnitDeletedTrap (1.3.6.1.3.94.0.3)

A connUnit has been deleted from this agent. The recommended severity level (for filtering) is “warning”. Sent whenever an Eport.OperChange event occurs and the connUnitTable is smaller than previously noted (A connUnit has gone away).

Variables: { connUnitId }

connUnitEventTrap (1.3.6.1.3.94.0.4)

An event has been generated by the connectivity unit. The recommended severity level (for filtering) is “info”. Sent when a change notification occurs that does not fit into any other specific category.

Variables:

{ connUnitEventId, connUnitEventType, connUnitEventObject, connUnitEventDescr }

Figure 2 provides the standard format of the connUnitEventDescr variable. Chassis, Blade, and Port are always 0.

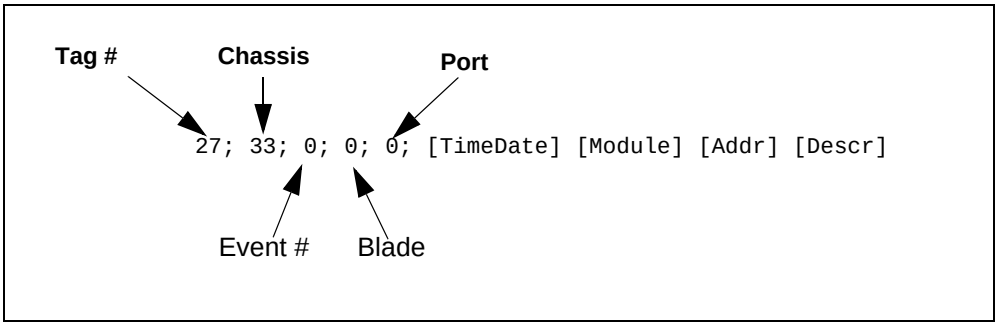

Figure 2. connUnitEventDescr variable format

Table 28 lists the fields in the connUnitEventDescr variable.

Table 28. connUnitEventDescr variable field descriptions

connUnitEventDescr variable	Description
Tag #	The number that identifies the event.
Event #	The event counter.
Chassis	The switch on which the event occurred.
Blade	The I/O blade on which the event occurred.

Table 28. connUnitEventDescr variable field descriptions (Continued)

connUnitEventDescr variable	Description
Port	The port on which the event occurred.
TimeDate	The time stamp of the event.
Module	The software module where the event was initiated.
Addr	The address in the software module where the event was initiated.
Descr	The description of the event.

Table 29 lists the trap information returned for the connUnitEventDescr variable.

Table 29. Filter trap levels

Trap type	Cause	Filter level
connUnitPortStatusChange	User port config change	eventSeverity_info
	User port state change	eventSeverity_info
	E_Port alarm	eventSeverity_critical
connUnitDeletedTrap	Fabric change and unit deleted	eventSeverity_info
connUnitStatusChange	Switch state change	eventSeverity_info
	Switch reset	eventSeverity_critical
connUnitSensorStatusChange	Power supply bad alarm	eventSeverity_critical
	Power supply OK alarm	eventSeverity_critical
	Fan bad alarm	eventSeverity_critical
	Fan OK alarm	eventSeverity_critical
	Overheat alarm	eventSeverity_critical
	Overwarm alarm	eventSeverity_critical
	Temperature OK alarm	eventSeverity_critical
connUnitEventTrap	SNMP config change	eventSeverity_info
	Switch config change	eventSeverity_info
	System config change	eventSeverity_info
	Topology change	eventSeverity_info
	Zoning change	eventSeverity_info
	Zoning merge failure	eventSeverity_critical
	NameServer change	eventSeverity_info
	Generic alarm	eventSeverity_critical
	Generic event	eventSeverity_warning

connUnitSensorStatusChange (1.3.6.1.3.94.0.5)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever any of the following notifications occur:

Chassis.PsBadAlarm

Chassis.PsOkAlarm

Chassis.FanBadAlarm

Chassis.FanOkAlarm

Blade.OverheatAlarm

Blade.OverwarmAlarm

Variables: { connUnitSensorStatus }

connUnitPortStatusChange (1.3.6.1.3.94.0.6)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever a UserPort.StateChange or UserPort.OperChange event occurs.

Enterprise: fcmgmt

Variables: { connUnitPortStatus, connUnitPortState }

coldStart

A coldStart trap signifies that the SNMPv2 entity, acting in an agent role, is re-initializing itself and that its configuration may have been altered.

authenticationFailure

An authenticationFailure trap signifies that the SNMPv2 entity, acting in an agent role, has received a protocol message that is not properly authenticated. While all implementations of the SNMPv2 must be capable of generating this trap, the snmpEnableAuthenTraps object indicates whether this trap will be generated.

Chapter 6. Fabric Element MIB objects

This chapter covers the implementation details for the Fabric Element Management Information Bases (FE-MIB) on the IBM FC3171 switch.

Fibre Channel FE MIB definitions

The textual substitutions in Table 30 are specific to the FE-MIB and can be used in place of primitive data types.

Table 30. FA-MIB textual substitutions

Description	Syntax
MilliSeconds	Unsigned32
MicroSeconds	Unsigned32
FcNameId	OCTET STRING (SIZE (8))
FcAddressId	OCTET STRING (SIZE (3))
FcRxDataFieldSize	Integer32 (128..2112)
FcBbCredit	Integer32 (0..32767)
FcphVersion	Integer32 (0..255)

FcStackedConnMode	INTEGER {
	none(1),
	transparent(2),
	lockedDown(3)
	}

FcCosCap	BITS {
	classF(0),
	class1(1),
	class2(2),
	class3(3),
	class4(4),
	class5(5),
	class6(6)
	}

Table 30. FA-MIB textual substitutions

Description	Syntax

FcCosCap	BITS {
	classF(0),
	class1(1),
	class2(2),
	class3(3),
	class4(4),
	class5(5),
	class6(6)
	}

FcFeModuleCapacity	Unsigned32
FcFeFxPortCapacity	Unsigned32
FcFeModuleIndex	Unsigned32
FcFeFxPortIndex	Unsigned32
FcFeNxPortIndex	Integer32 (1..126)
FcBbCreditModel	INTEGER { regular(1), alternate (2) }

Configuration group

This group consists of scalar objects and tables. It contains the configuration and service parameters of the Fabric Element and the FxPorts. The group represents a set of parameters associated with the Fabric Element or an FxPort to support its NxPorts. The objects described in this section are not in a table format. An example of how to access one of these objects is:

```
"snmpget localhost public fcFeFabricName.0".
```

fcFeFabricName (1.3.6.1.2.1.75.1.1.1)

The Name_Identifier of the Fabric to which this Fabric Element belongs.

Syntax FcNameId

Access read-write

Status Current

Return Value World Wide Name of the principal switch. For example, 10 00 00 C0 DD 00 71 C2. Writes are not supported.

fcFeElementName (1.3.6.1.2.1.75.1.1.2)

The Name_Identifier of the Fabric Element.

Syntax FcNameId

Access read-write

Status Current

Return Value World Wide Name of the switch. For example, 10 00 00 C0 DD 00 71 C9. Writes are not supported.

fcFeModuleCapacity (1.3.6.1.2.1.75.1.1.3)

The maximum number of modules in the Fabric Element, regardless of their current state.

Syntax FcFeModuleCapacity

Access read-only

Status Current

Return Value Total number of switches in the fabric if ProxyEnable setting is Enabled on the out-of-band switch. If ProxyEnable setting is disabled on the out-of-band switch, return value = 1.

Module table

The objects described in this section are in table format indexed by switch. An example of how to access one of these objects is: "snmpget localhost public fcFeModuleDescr.1". This table contains one entry for each module.

fcFeModuleDescr (1.3.6.1.2.1.75.1.1.4.1.2)

A textual description of the module. This value should include the full name and version identification of the module.

Syntax SnmpAdminString

Access read-only

Status current

Return Value `IBM Flex System FC3171 8 Gb SAN Switch` or `IBM Flex System FC3171 8 Gb Pass-thru`.

fcFeModuleObjectID (1.3.6.1.2.1.75.1.1.4.1.3)

The vendor's authoritative identification of the module. This value may be allocated within the SMI enterprises subtree (1.3.6.1.4.1), and provides a means for determining what kind of module is being managed.

For example, this object could take the value 1.3.6.1.4.1.99649.3.9 if vendor "Neufe Inc." was assigned the subtree 1.3.6.1.4.1.99649, and had assigned the identifier 1.3.6.1.4.1.99649.3.9 to its FeFiFo-16 PlugInCard.

Syntax OBJECT IDENTIFIER

Access read-only

Status current

Return Value `1.3.6.1.4.1.3873.1.33`.

fcFeModuleOperStatus (1.3.6.1.2.1.75.1.1.4.1.4)

Switch definitions map 1-to-1 with the MIB definitions. This object indicates the operational status of the module.

- online (1) - the module is functioning properly
- offline (2) - the module is not available
- testing (3) - the module is under testing
- faulty (4) - the module is defective in some way

Syntax

INTEGER {
online(1), - functional
offline(2), - not available
testing(3), - under testing
faulty(4) - defective
}

Access

read-only

Status

Current

Return Value

Table 31 lists the module operational status.

Table 31. Module operational status return values

Mode	Return value
online	online(1)
offline	offline(2)
diagnostics	testing(3)
other	faulty(4)

fcFeModuleLastChange (1.3.6.1.2.1.75.1.1.4.1.5)

This object contains the value of sysUpTime when the module entered its current operational status. A value of zero indicates that the operational status of the module has not changed since the agent last restarted.

Syntax TimeStamp

Access read-only

Status Current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFeModuleFxPortCapacity (1.3.6.1.2.1.75.1.1.4.1.6)

The number of FxPort that can be contained within the module. Within each module, the ports are uniquely numbered in the range from 1 to fcFeModuleFxPortCapacity inclusive. However, the numbers are not required to be contiguous.

Syntax FcFeFxPortCapacity

Access read-only

Status current

Return Value 20

fcFeModuleName (1.3.6.1.2.1.75.1.1.4.1.7)

The Name_Identifier of the switch.

Syntax FcNameId

Access read-write

Status current

Return Value World Wide Name of the switch. Writes are not supported. For example, 10 00 00 C0 DD 00 71 C9.

FxPort configuration table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: snmpget localhost public fcFxPortName.1.1. This table contains one entry for each FxPort and Configuration parameters of the ports.

fcFxPortName (1.3.6.1.2.1.75.1.1.5.1.2)

The World Wide Name of this FxPort. Each FxPort has a unique Port World Wide Name within the Fabric.

Syntax FcNameId

Access read-only

Status current

Return Value Port World Wide Name for each port on switch. For example, the return value for port #2 would be 20 02 00 C0 DD 00 71 C9, and the return value for port #14 would be 20 0E 00 C0 DD 00 71 C9.

fcFxPortFcphVersionHigh (1.3.6.1.2.1.75.1.1.5.1.3)

The highest or most recent version of FC-PH that the FxPort is configured to support.

Syntax FcphVersion

Access read-only

Status Current

Return Value 32 (0x20).

fcFxPortFcphVersionLow (1.3.6.1.2.1.75.1.1.5.1.4)

The lowest or earliest version of FC-PH that the FxPort is configured to support.

Syntax FcphVersion

Access read-only

Status current

Return Value 9

fcFxPortBbCredit (1.3.6.1.2.1.75.1.1.5.1.5)

The total number of receive buffers available for holding Class 1 connect-request, Class 2, or Class3 frames from the attached NxPort. It is for buffer-to-buffer flow control in the direction from the attached NxPort (if applicable) to FxPort.

Syntax FcBbCredit

Access read-only

Status current

Return Value 16

fcFxPortRxBufSize (1.3.6.1.2.1.75.1.1.5.1.6)

The largest Data_Field Size (in octets) for an FT_1 frame that can be received by the FxPort.

Syntax FcRxDataFieldSize

Access read-only

Status current

Return Value 2112 (0x840).

fcFxPortRatov (1.3.6.1.2.1.75.1.1.5.1.7)

The Resource_Allocation_Timeout Value configured for the FxPort. This is used as the timeout value for determining when to reuse an NxPort resource such as a Recovery_Qualifier. It represents E_D_TOV plus twice the maximum time that a frame may be delayed within the fabric and still be delivered. Refer to "fcFxPortEdtov (1.3.6.1.2.1.75.1.1.5.1.8)" on page 6-431 for more information.

Syntax MilliSeconds

Access read-only

Status Current

Return Value Default: 10000 (0x2710).

fcFxPortEdtov (1.3.6.1.2.1.75.1.1.5.1.8)

The E_D_TOV value configured for the FxPort. The Error_Detect_Timeout Value is used as the timeout value for detecting an error condition.

Syntax MilliSeconds

Access read-only

Status current

Return Value Default: 2000 (0x7D0).

fcFxPortCosSupported (1.3.6.1.2.1.75.1.1.5.1.9)

A value indicating the set of classes of service supported by the FxPort.

Syntax FcCosCap

Access read-only

Status Current

Return Value Class 3, 2, and F (0x0D).

fcFxPortIntermixSupported (1.3.6.1.2.1.75.1.1.5.1.10)

A flag indicating whether or not the FxPort supports an Intermixed Dedicated Connection.

Syntax TruthValue

Access read-only

Status current

Return Value False (2).

fcFxPortStackedConnMode (1.3.6.1.2.1.75.1.1.5.1.11)

A value indicating the mode of Stacked Connect supported by the FxPort.

Syntax FcStackedConnMode

Access read-only

Status current

Return Value None (1).

fcFxPortClass2SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.12)

A flag indicating whether or not Class 2 Sequential Delivery is supported by the FxPort.

Syntax TruthValue

Access read-only

Status current

Return Value True (1).

fcFxPortClass3SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.13)

A flag indicating whether or not Class 3 Sequential Delivery is supported by the FxPort.

Syntax TruthValue

Access read-only

Status current

Return Value True (1).

fcFxPortHoldTime (1.3.6.1.2.1.75.1.1.5.1.14)

The maximum time, in microseconds, that the FxPort shall hold a frame before discarding the frame if it is unable to deliver the frame. The value 0 means that the FxPort does not support this parameter.

Syntax MicroSeconds

Access read-only

Status current

Return Value Default ED_TOV parameter: 2000 (0x7D0).

The Status group

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortId.1.1". This group consists of tables that contain operational status and established service parameters for the Fabric Element and the attached NxPorts.

This table contains one entry for each FxPort, and the operational status and parameters of the FxPorts.

fcFxPortID (1.3.6.1.2.1.75.1.2.1.1.1)

The address identifier by which this FxPort is identified within the fabric. The FxPort may assign its address identifier to its attached NxPort(s) during Fabric Login.

Syntax FcAddressId

Access read-only

Status current

Return Value Address of each port based on Domain, Area, and ALPA. Example, 64 03 00.

fcFxPortBbCreditAvailable (1.3.6.1.2.1.75.1.2.1.1.2)

The number of buffers currently available for receiving frames from the attached port in the buffer-to-buffer flow control. The value should be less than or equal to fcFxPortBbCredit.

Syntax Gauge32

Access read-only

Status Current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortOperMode (1.3.6.1.2.1.75.1.2.1.1.3)

The current operational mode of the FxPort.

Syntax INTEGER { unknown(1), fPort(2), flPort(3) }

Access read-only

Status current

Return Value Table 32 lists the fcFxPortOperMode return values.

Table 32. Port operational modes

Mode	Return value
Unknown	1
F_Port	2
FL_Port	3

fcFxPortAdminMode (1.3.6.1.2.1.75.1.2.1.1.4)

The desired operational mode of the FxPort.

Syntax INTEGER { fPort(2), flPort(3) }

Access read-write

Status Current

Return Value Unsupported. Always returns error status `NoSuchName`.

FxPort physical level table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortPhysAdminStatus.1.1".

This table contains one entry for each FxPort in the Fabric Element, and the physical level status, and parameters of the FxPorts.

fcFxPortPhysAdminStatus (1.3.6.1.2.1.75.1.2.2.1.1)

The desired state of the FxPort. A management station may place the FxPort in a desired state by setting this object accordingly. The testing(3) state indicates that no operational frames can be passed. When a Fabric Element initializes, all FxPorts start with fcFxPortPhysAdminStatus in the offline(2) state. As the result of either explicit management action or per configuration information accessible by the Fabric Element, fcFxPortPhysAdminStatus is then changed to either the online(1) or testing(3) states, or remains in the offline state.

Syntax

INTEGER {
online(1), - place port online
offline(2), - take port offline
testing(3) - initiate test procedures
}

Access

read-write

Status

current

Return Value

Table 33 lists the fcFxPortPhysAdminStatus read values.

Table 33. fcFxPortPhysAdminStatus read return values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)
Down	offline (2)

Table 34 lists the fcFxPortPhysAdminStatus write values.

Table 34. fcFxPortPhysAdminStatus write values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)

fcFxPortPhysOperStatus (1.3.6.1.2.1.75.1.2.2.1.2)

The current operational status of the FxPort. The testing(3) indicates that no operational frames can be passed. If fcFxPortPhysAdminStatus is offline(2), then fcFxPortPhysOperStatus should be offline(2). If fcFxPortPhysAdminStatus is changed to online(1), then fcFxPortPhysOperStatus should change to online(1). If the FxPort is ready to accept Fabric Login request from the attached NxPort, it should proceed and remain in the link- failure(4) state if, and only if, there is a fault that prevents it from going to the online(1) state.

Syntax

INTEGER {
online(1), - Login may proceed
offline(2), - Login cannot proceed
testing(3), - port is under test
linkFailure(4) - failure after online/testing
}

Access

read-only

Status

current

Return Value

Table 35 lists the fcFxPortPhysOperStatus return values.

Table 35. fcFxPortPHysOperStatus return values

Status	Return value
Online	online (1)
Offline	offline (2)
Diagnostic	testing (3)
Down	linkfailure (4)

fcFxPortPhysLastChange (1.3.6.1.2.1.75.1.2.2.1.3)

The value of sysUpTime at the time the FxPort entered its current operational status. A value of zero indicates that the FxPort's operational status has not changed since the agent last restarted.

Syntax TimeStamp

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortPhysRttov (1.3.6.1.2.1.75.1.2.2.1.4)

The Receiver_Transmitter_Timeout value of the FxPort. This is used by the receiver logic to detect a loss of synchronization.

Syntax Milliseconds

Access read-write

Status current

Return Value Default RT_TOV parameter: 100 (0x64). This is a global setting for the switch. If writing value to a port, all ports will reflect this new value.

Fx Port fabric login table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortFcphVersionAgreed.1.1". This table contains one entry for each FxPort in the fabric element and the service parameters that have been established from the most recent Fabric Login (implicit or explicit).

This table contains one entry for each FxPort in the fabric element, and the service parameters that have been established from the most recent Fabric Login, implicit or explicit.

fcFxPortFcphVersionAgreed (1.3.6.1.2.1.75.1.2.3.1.2)

The version of FC-PH that the FxPort has agreed to support from the Fabric Login.

Syntax FcphVersion

Access read-only

Status current

Return Value Unsupported

fcFxPortNxPortBbCredit (1.3.6.1.2.1.75.1.2.3.1.3)

The total number of buffers available for holding class 1 connect-request, class 2, or class 3 frames to be transmitted to the attached NxPort. It is for buffer-to-buffer flow control in the direction from FxPort to NxPort. The buffer-to-buffer flow control mechanism is indicated in the respective fcFxPortBbCreditModel.

Syntax FcBbCredit

Access read-only

Status current

Return Value Unsupported

fcFxPortNxPortRxDataFieldSize (1.3.6.1.2.1.75.1.2.3.1.4)

The Receive Data Field Size of the attached NxPort. This object specifies the largest Data Field Size for an FT_1 frame that can be received by the NxPort.

Syntax FcRxDataFieldSize

Access read-only

Status current

Return Value Unsupported

fcFxPortCosSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.5)

A variable indicating that the attached NxPort has requested the FxPort for the support of classes of services and the FxPort has granted the request.

Syntax FcCosCap

Access read-only

Status current

Return Value The bits have the following bit-mapped definition:

Bit 7 Class-six
Bit 6 Class-five
Bit 5 Class-four
Bit 4 Class-three
Bit 3 Class-two
Bit 2 Class-one
Bit 1 Class F

For example: If Class 3, return value 0x10.

fcFxPortIntermixSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.6)

A variable indicating that the attached NxPort has requested the FxPort for the support of Intermix and the FxPort has granted the request. This flag is only valid if Class 1 service is supported.

Syntax TruthValue

Access read-only

Status current

Return Value False (2)

fcFxPortStackedConnModeAgreed (1.3.6.1.2.1.75.1.2.3.1.7)

A variable indicating whether the FxPort has agreed to support stacked connect from the Fabric Login. This is only meaningful if the ports are using Class 1 service.

Syntax FcStackedConnMode

Access read-only

Status current

Return Value None (1)

fcFxPortClass2SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.8)

A variable indicating whether the FxPort has agreed to support Class 2 sequential delivery from the Fabric Login. This is only meaningful if the ports are using Class 2 service.

Syntax TruthValue

Access read-only

Status Current

Return Value True (1)

fcFxPortClass3SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.9)

A flag indicating whether the FxPort has agreed to support Class 3 sequential delivery from the Fabric Login. This is only meaningful if the ports are using Class 3 service.

Syntax TruthValue

Access read-only

Status current

Return Value True (1)

fcFxPortNxPortName (1.3.6.1.2.1.75.1.2.3.1.10)

The port name of the attached NxPort.

Syntax FcNameId

Access read-only

Status Current

Return Value Switch port's Port World Wide Name for the attached device

fcFxPortConnectedNxPort (1.3.6.1.2.1.75.1.2.3.1.11)

The address identifier of the destination NxPort with which this FxPort is currently engaged in a either a Class 1 or loop connection. If this FxPort is not engaged in a connection, then the value of this object is “000000”H.

Syntax FcAddressId

Access read-only

Status Current

Return Value Unsupported

fcFxPortBbCreditModel (1.3.6.1.2.1.75.1.2.3.1.12)

This object identifies the BB_Credit model used by the FxPort.

Syntax FcBbCreditModel

Access read-write

Status current

Return Value Alternate (2). Writes not supported.

The Error group

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortLinkFailures.1.1". This group consists of tables that contain information about the various types of errors detected. The management station may use the information in this group to determine the quality of the link between the FxPort and its attached NxPort.

The FxPort Error table contains, one entry for each FxPort in the Fabric Element, counters recording numbers of errors detected since the management agent re-initialized. The first 6 columnar objects after the port index corresponds to the counters in the Link Error Status Block.

fcFxPortLinkFailures (1.3.6.1.2.1.75.1.3.1.1.1)

The number of link failures detected by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of LinkFailures encountered for a port

fcFxPortSyncLosses (1.3.6.1.2.1.75.1.3.1.1.2)

The number of loss of synchronizations detected by the FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of LossOfSyncs encountered for a port

fcFxPortSigLosses (1.3.6.1.2.1.75.1.3.1.1.3)

The number of loss of signals detected by the FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortPrimSeqProtoErrors (1.3.6.1.2.1.75.1.3.1.1.4)

The number of primitive sequence protocol errors detected by the FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of PrimSeqErrors encountered for a port

fcFxPortInvalidTxWords (1.3.6.1.2.1.75.1.3.1.1.5)

The number of invalid transmission words detected by the FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of DecodeErrors encountered for a port

fcFxPortInvalidCrcs (1.3.6.1.2.1.75.1.3.1.1.6)

The number of invalid CRCs detected by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of InvalidCRCs encountered for a port

fcFxPortDelimiterErrors (1.3.6.1.2.1.75.1.3.1.1.7)

The number of Delimiter Errors detected by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortAddressIdErrors (1.3.6.1.2.1.75.1.3.1.1.8)

The number of address identifier errors detected by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of InvDestAddrs encountered for a port

fcFxPortLinkResetIns (1.3.6.1.2.1.75.1.3.1.1.9)

The number of Link Reset Protocols received by this FxPort from the attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of RxLinkResets received by a port

fcFxPortLinkResetOuts (1.3.6.1.2.1.75.1.3.1.1.10)

The number of Link Reset Protocols issued by this FxPort to the attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of TxLinkResets sent by a port

fcFxPortOlsIns (1.3.6.1.2.1.75.1.3.1.1.11)

The number of Offline Sequences received by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of RxOfflineSeqs received by a port

fcFxPortOlsOuts (1.3.6.1.2.1.75.1.3.1.1.12)

The number of Offline Sequences issued by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of TxOflineSeqs sent by a port

Class 1 accounting group

The class 1 accounting group consists of a table that contains information for the Fx ports in the Fabric Element. The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC1InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent was re-initialized.

fcFxPortC1InFrames (1.3.6.1.2.1.75.1.4.1.1.1)

The number of Class 1 frames (other than Class 1 connect-request) received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1OutFrames (1.3.6.1.2.1.75.1.4.1.1.2)

The number of Class 1 frames (other than Class 1 connect- request) delivered through this FxPort to its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1InOctets (1.3.6.1.2.1.75.1.4.1.1.3)

The number of Class 1 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1OutOctets (1.3.6.1.2.1.75.1.4.1.1.4)

The number of Class 1 frame octets, including the frame delimiters, delivered through this FxPort its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1Discards (1.3.6.1.2.1.75.1.4.1.1.5)

The number of Class 1 frames discarded by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1FbsyFrames (1.3.6.1.2.1.75.1.4.1.1.6)

The number of F_BSY frames generated by this FxPort against Class 1 connect-request.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1FrjtFrames (1.3.6.1.2.1.75.1.4.1.1.7)

The number of F_RJT frames generated by this FxPort against Class 1 connect-request.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1InConnections (1.3.6.1.2.1.75.1.4.1.1.8)

The number of Class 1 connections successfully established in which the attached NxPort is the source of the connect-request.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1OutConnections (1.3.6.1.2.1.75.1.4.1.1.9)

The number of Class 1 connections successfully established in which the attached NxPort is the destination of the connect-request.

Syntax Counter32

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

fcFxPortC1ConnTime (1.3.6.1.2.1.75.1.4.1.1.10)

The cumulative time that this FxPort has been engaged in Class 1 connection. The amount of time is counted from after a connect-request has been accepted until the connection is disengaged, either by an EOFdt or Link Reset.

Syntax Milliseconds

Access read-only

Status current

Return Value Unsupported. Always returns error status `NoSuchName`.

Class 2 accounting group

The class 2 accounting group consists of a table that contains information for the Fx ports in the Fabric Element.The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC2InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent was re-initialized.

fcFxPortC2InFrames (1.3.6.1.2.1.75.1.4.2.1.1)

The number of Class 2 frames received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class2FramesIn received by a port

fcFxPortC2OutFrames (1.3.6.1.2.1.75.1.4.2.1.2)

The number of Class 2 frames delivered through this FxPort to its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class2FramesOut sent by a port

fcFxPortC2InOctets (1.3.6.1.2.1.75.1.4.2.1.3)

The number of Class 2 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class2WordsIn received by a port

fcFxPortC2OutOctets (1.3.6.1.2.1.75.1.4.2.1.4)

The number of Class 2 frame octets, including the frame delimiters, delivered through this FxPort to its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class2WordsOut sent by a port

fcFxPortC2Discards (1.3.6.1.2.1.75.1.4.2.1.5)

The number of Class 2 frames discarded by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class2Toss discarded by a port

fcFxPortC2FbsyFrames (1.3.6.1.2.1.75.1.4.2.1.6)

The number of F_BSY frames generated by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of FBusy frames generated by this port for Class 2 and 3 frames

fcFxPortC2FrjtFrames (1.3.6.1.2.1.75.1.4.2.1.7)

The number of F_RJT frames generated by this FxPort against Class 2 frames.

Syntax Counter32

Access read-only

Status current

Return Value Total number of FReject frames generated by this port for Class 2 and 3 frames

Class 3 accounting group

The class 3 accounting group consists of a table that contains information for the Fx ports in the Fabric Element.The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC3InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent has re-initialized.

fcFxPortC3InFrames (1.3.6.1.2.1.75.1.4.3.1.1)

The number of Class 3 frames received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class3FramesIn received by a port

fcFxPortC3OutFrames (1.3.6.1.2.1.75.1.4.3.1.2)

The number of Class 3 frames delivered through this FxPort to its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class3FramesOut sent by a port

fcFxPortC3InOctets (1.3.6.1.2.1.75.1.4.3.1.3)

The number of Class 3 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class3WordsOut received by a port

fcFxPortC3OutOctets (1.3.6.1.2.1.75.1.4.3.1.4)

The number of Class 3 frame octets, including the frame delimiters, delivered through this FxPort to its attached NxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class3WordsOut sent by a port

fcFxPortC3Discards (1.3.6.1.2.1.75.1.4.3.1.5)

The number of Class 3 frames discarded by this FxPort.

Syntax Counter32

Access read-only

Status current

Return Value Total number of Class3Toss discarded by a port

Capability group

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortName.1.1". The Capability Group consists of a table describing information about what each FxPort is inherently capable of operating or supporting. A capability may be used as expressed in its respective object value in the Configuration group.

fcFxPortCapFcphVersionHigh (1.3.6.1.2.1.75.1.5.1.1.1)

The highest or most recent version of FC-PH that the FxPort is capable of supporting.

Syntax FcphVersion

Access read-only

Status current

Return Value 32 (0x20)

fcFxPortCapFcphVersionLow (1.3.6.1.2.1.75.1.5.1.1.2)

The lowest or earliest version of FC-PH that the FxPort is capable of supporting.

Syntax FcphVersion

Access read-only

Status current

Return Value 9

fcFxPortCapBbCreditMax (1.3.6.1.2.1.75.1.5.1.1.3)

The maximum number of receive buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames from the attached NxPort.

Syntax FcBbCredit

Access read-only

Status current

Return Value Default: 255 (0xFF)

fcFxPortCapBbCreditMin (1.3.6.1.2.1.75.1.5.1.1.4)

The minimum number of receive buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames from the attached NxPort.

Syntax FcBbCredit

Access read-only

Status current

Return Value Default is: 0 (0x00)

fcFxPortCapRxDataFieldSizeMax (1.3.6.1.2.1.75.1.5.1.1.5)

The maximum size in bytes of the Data Field in a frame that the FxPort is capable of receiving from its attached NxPort.

Syntax FcRxDataFieldSize

Access read-only

Status current

Return Value 2112 (0x840)

fcFxPortCapRxDataFieldSizeMin (1.3.6.1.2.1.75.1.5.1.1.6)

The minimum size in bytes of the Data Field in a frame that the FxPort is capable of receiving from its attached NxPort.

Syntax FcRxDataFieldSize

Access read-only

Status current

Return Value 128 (0x80)

fcFxPortCapCos (1.3.6.1.2.1.75.1.5.1.1.7)

A value indicating the set of classes of service that the FxPort is capable of supporting.

Syntax FcCosCap

Access read-only

Status current

Return Value Class F, 2, and 3 (0x0d)

fcFxPortCapIntermix (1.3.6.1.2.1.75.1.5.1.1.8)

A flag indicating whether or not the FxPort is capable of supporting the intermixing of Class 2 and Class 3 frames during a Class 1 connection. This flag is only valid if the port is capable of supporting Class 1 service.

Syntax TruthValue

Access read-only

Status current

Return Value False (2)

fcFxPortCapStackedConnMode (1.3.6.1.2.1.75.1.5.1.1.9)

A value indicating the mode of Stacked Connect request that the FxPort is capable of supporting.

Syntax FcStackedConnMode

Access read-only

Status current

Return Value None (1)

fcFxPortCapClass2SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.10)

A flag indicating whether or not the FxPort is capable of supporting Class 2 Sequential Delivery.

Syntax TruthValue

Access read-only

Status current

Return Value True (1)

fcFxPortCapClass3SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.11)

A flag indicating whether or not the FxPort is capable of supporting Class 3 Sequential Delivery.

Syntax TruthValue

Access read-only

Status current

Return Value True (1)

fcFxPortCapHoldTimeMaxv (1.3.6.1.2.1.75.1.5.1.1.12)

The maximum holding time that the FxPort is capable of supporting, in microseconds.

Syntax MicroSeconds

Access read-only

Status current

Return Value 20000 (0x4E20)

fcFxPortCapHoldTimeMin (1.3.6.1.2.1.75.1.5.1.1.13)

The minimum holding time that the FxPort is capable of supporting, in microseconds.

Syntax MicroSeconds

Access read-only

Status current

Return Value 10 (0x0A)

Chapter 7. Private enterprise MIB objects

This chapter covers the implementation details for the Private (Enterprise-specific) Management Information Bases on the IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru.

Private Enterprise MIB definitions

This MIB replaces the fcFxPortPhysTable module defined in FIBRE-CHANNEL-FE-MIB, and defines volatile control objects for ports in a the IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru. If the switch gets reset, these values revert back to the default values in the configuration file.

fcQxPortPhysAdminStatus (1.3.6.1.4.1.1663.1.3.10.1.1.3)

The desired state of the FxPort. A management station may place the FxPort in a desired state by setting this object accordingly. The testing(3) state indicates that no operational frames can be passed. When a Fabric Element initializes, all FxPorts start with fcQxPortPhysAdminStatus in the offline(2) state. As the result of either explicit management action or per configuration information accessible by the Fabric Element, fcQxPortPhysAdminStatus is then changed to either the online(1) or testing(3) states, or remains in the offline state.

Syntax

INTEGER {
online(1), - place port online
offline(2), - take port offline
testing(3) - initiate test procedures
}

Access

read-write

Status

current

Return Value

Table 36 lists the fcQxPortPhysAdminStatus read values.

Table 36. fcQxPortPhysAdminStatus read return values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)
Down	offline (2)

Table 37 lists the fcQxPortPhysAdminStatus write values.

Table 37. fcQxPortPhysAdminStatus write values

Port	**Value**
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)

fcQxPortPhysOperStatus (1.3.6.1.4.1.1663.1.3.10.1.1.4)

The current operational status of the FxPort. The testing(3) indicates that no operational frames can be passed. If fcQxPortPhysAdminStatus is offline(2), then fcQxPortPhysOperStatus should be offline(2). If fcQxPortPhysAdminStatus is changed to online(1), then fcQxPortPhysOperStatus should change to online(1). If the FxPort is ready to accept Fabric Login request from the attached NxPort, it should proceed and remain in the link- failure(4) state if, and only if, there is a fault that prevents it from going to the online(1) state.

Syntax

INTEGER {
online(1), - Login may proceed
offline(2), - Login cannot proceed
testing(3), - port is under test
linkFailure(4) - failure after online/testing
}

Access

read-only

Status

current

Return Value

Table 38 lists the fcQxPortPhysOperStatus return values.

Table 38. fcFxPortPHysOperStatus return values

Status	Return value
Online	online (1)
Offline	offline (2)
Diagnostic	testing (3)
Down	linkfailure (4)

Related Traps

The following traps contain the trap information being sent from the agent to the manager.

qlSB2PortLinkDown (qLogicExperimental 0 10)

A linkDown trap signifies that the SNMP entity, acting in an agent role, has detected that the fcQxPortPhysOperStatus object for one of its communication links has left the online state and transitioned to some other state. The current state is indicated by the included value of fcQxPortPhysOperStatus.

Variables:

{ fcQxPortPhysAdminStatus, fcQxPortPhysOperStatus }

qlSB2PortLinkUp (qLogicExperimental 0 11)

A linkUp trap signifies that the SNMP entity, acting in an agent role, has detected that the fcQxPortPhysOperStatus object for one of its communication links has entered the online state from some other state. The current state is indicated by the included value of fcQxPortPhysOperStatus.

Variables:

{ fcQxPortPhysAdminStatus, fcQxPortPhysOperStatus }

qlconnUnitAddedTrap (qLogicExperimental 0 12)

A connUnit has been aded to this agent.

Variables:

{ connUnitId }

Chapter 8. Firmware download MIB objects

This chapter covers the implementation details for the Firmware Download Management Information Bases (FD-MIB) on the IBM FC3171 switch.

Firmware download MIB definitions

The FD-MIB enables you to download, install, and activate new firmware on an IBM FC3171 switch using the Trivial File Transfer Protocol (TFTP). The downloaded firmware can be activated using a hot reset (non-disruptive) or a hard reset (disruptive).

A hot reset is a Non-Disruptive Code Load Activation (NDCLA) operation. The firmware will be activated, the switch will be reset without a Power On Self Test, and switch traffic will not be disrupted. The switch does not need to be rebooted after the firmware is activated. During a hot reset operation, fabric services will be unavailable for a short period (30–75 seconds depending on switch model). To ensure that the NDCLA operation is successful, verify that all administrative changes to the fabric (if any) are complete. When you need to do NDCLA/hot reset to multiple switches, only perform the NDCLA/hot reset on one switch at a time, and wait 75 seconds before performing the NDCLA/hot reset operation on the next switch.

A hard reset is a Normal (or regular) reset operation. The firmware will be activated, the switch will be reset with a Power On Self Test, and switch traffic will be disrupted. The switch must be rebooted after the firmware is activated.

qlgcChFwOpResult (1.3.6.1.4.1.3873.3.1.1.2.1)

The status of the last firmware download and/or installation attempt.

Syntax OBJECT IDENTIFIER

Access read-only

Status Current

Return Value Returns the following:

DownloadNoError: 1.3.6.1.4.1.3873.3.1.1.1.1.1,
DownloadHostError: 1.3.6.1.4.1.3873.3.1.1.1.1.2,
DownloadFileError: 1.3.6.1.4.1.3873.3.1.1.1.1.3,
InstallNoError: 1.3.6.1.4.1.3873.3.1.1.1.2.1,
InstallFileError: 1.3.6.1.4.1.3873.3.1.1.1.2.2,
InstallFileErrorNoAdmin: 1.3.6.1.4.1.3873.3.1.1.1.2.3,
ResetNoError: 1.3.6.1.4.1.3873.3.1.1.1.3.1,
ResetNoErr: 1.3.6.1.4.1.3873.3.1.1.1.3.2,
ResetNoAdmin: 1.3.6.1.4.1.3873.3.1.1.1.3.3

qlgcChFwOpRequest (1.3.6.1.4.1.3873.3.1.1.2.2)

Starts the operation to download/install firmware, and/or reset the switch.

Syntax

INTEGER {
(1) - auto (downloads, installs, and resets the switch)
(2) - downloadOnly
(3) - installOnly
(4) - resetOnly
}

Access

read-write

Status

current

qlgcChFwDwldHostAddrType (1.3.6.1.4.1.3873.3.1.1.2.3)

The type of the IP address from which the firmware file is accessed.

Syntax INTEGER

Access read-write

Status current

qlgcChFwDwldHostAddr (1.3.6.1.4.1.3873.3.1.1.2.4)

The IP address from which the firmware file is accessed.

Syntax IP ADDRESS

Access read-write

Status current

qlgcChFwDwldHostPort (1.3.6.1.4.1.3873.3.1.1.2.5)

The port number (defaults is 69) used to transfer the firmware file.

Syntax INTEGER

Access read-write

Status current

qlgcChFwDwldPathName (1.3.6.1.4.1.3873.3.1.1.2.6)

The full directory name on the server where the firmware file is located. If the firmware file to be downloaded is in a subdirectory, setting this path name to the name of that subdirectory is required.

Syntax DisplayString

Access read-write

Status current

qlgcChFwDwldFileName (1.3.6.1.4.1.3873.3.1.1.2.7)

The filename of the firmware being transferred.

Syntax DisplayString

Access read-write

Status current

qlgcChFwResetMethod (1.3.6.1.4.1.3873.3.1.1.2.8)

The value for the type of reset (hot reset or hard reset).

Syntax

INTEGER {
(1) - Normal (disruptive)
(2) - NDCLA (Non-Disruptive Code Load Activation)

Access

read-write

Status

current

Appendix A. Getting help and technical assistance

If you need help, service, or technical assistance or just want more information about IBM products, you will find a wide variety of sources available from IBM to assist you. This section contains information about where to go for additional information about IBM and IBM products, what to do if you experience a problem with your system, and whom to call for service, if it is necessary.

Before you call

Before you call, make sure that you have taken these steps to try to solve the problem yourself:

- Check all cables to make sure that they are connected.
- Check the power switches to make sure that the system and any optional devices are turned on.
- Use the troubleshooting information in your system documentation, and use the diagnostic tools that come with your system.
- Go to the IBM support website at http://www.ibm.com/supportportal/ to check for technical information, hints, tips, and new device drivers or to submit a request for information.

You can solve many problems without outside assistance by following the troubleshooting procedures that IBM provides in the online help or in the documentation that is provided with your IBM product. The documentation that comes with IBM systems also describes the diagnostic tests that you can perform. Most systems, operating systems, and programs come with documentation that contains troubleshooting procedures and explanations of error messages and error codes. If you suspect a software problem, see the documentation for the operating system or program.

Using the documentation

Information about your IBM system and preinstalled software, if any, or optional device is available in the documentation that comes with the product. That documentation can include printed documents, online documents, readme files, and help files. See the troubleshooting information in your system documentation for instructions for using the diagnostic programs. The troubleshooting information or the diagnostic programs might tell you that you need additional or updated device drivers or other software. IBM maintains pages on the World Wide Web where you can get the latest technical information and download device drivers and updates. To access these pages, go to http://www.ibm.com/supportportal/ and follow the instructions.

Getting help and information from the World Wide Web

On the World Wide Web, the IBM website has up-to-date information about IBM systems, optional devices, services, and support. You can find service information for IBM systems and optional devices at http://www.ibm.com/supportportal/.

Software service and support

Through IBM Support Line, you can get telephone assistance, for a fee, with usage, configuration, and software problems. For information about which products are supported by Support Line in your country or region, see http://www.ibm.com/services/supline/products/.

For more information about Support Line and other IBM services, see http://www.ibm.com/services/, or see http://www.ibm.com/planetwide/ for support telephone numbers. In the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

Hardware service and support

You can receive hardware service through your IBM reseller or IBM Services. To locate a reseller authorized by IBM to provide warranty service, go to http://www.ibm.com/partnerworld/ and click **Find Business Partners** on the right side of the page. For IBM support telephone numbers, see http://www.ibm.com/planetwide/. In the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

In the U.S. and Canada, hardware service and support is available 24 hours a day, 7 days a week. In the U.K., these services are available Monday through Friday, from 9 a.m. to 6 p.m.

IBM Taiwan product service

台灣 IBM 產品服務聯絡方式：
台灣國際商業機器股份有限公司
台北市松仁路 7 號 3 樓
電話：0800-016-888

IBM Taiwan product service contact information:
IBM Taiwan Corporation
3F, No 7, Song Ren Rd.
Taipei, Taiwan
Telephone: 0800-016-888

Appendix B. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product, and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Adobe and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom.

Intel, Intel Xeon, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Important notes

When referring to hard disk drive capacity or communications volume, MB stands for 1 000 000 bytes, and GB stands for 1 000 000 000 bytes. Total user-accessible capacity can vary depending on operating environments.

IBM makes no representation or warranties regarding non-IBM products and services that are ServerProven®, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. These products are offered and warranted solely by third parties.

IBM makes no representations or warranties with respect to non-IBM products. Support (if any) for the non-IBM products is provided by the third party, not IBM.

Some software might differ from its retail version (if available) and might not include user manuals or all program functionality.

This product is not intended to be connected directly or indirectly by any means whatsoever to interfaces of public telecommunications networks, nor is it intended to be used in a public services network.

Documentation format

The publications for this product are in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when you use the PDF files and want to request a web-based format or accessible PDF document for a publication, direct your mail to the following address:

Information Development
IBM Corporation
205/A015
3039 E. Cornwallis Road
P.O. Box 12195
Research Triangle Park, North Carolina 27709-2195
U.S.A.

In the request, be sure to include the publication part number and title.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Glossary

AL_PA
Arbitrated Loop Physical Address

Arbitrated Loop
A Fibre Channel topology where ports use arbitration to establish a point-to-point circuit.

Arbitrated Loop Physical Address (AL_PA)
A unique one-byte value assigned during loop initialization to each NL_Port on a Loop.

Abstract Syntax Notation (ASN.1)
Abstract Syntax Notation number One (ASN.1) is an international standard that specifies data used in communication protocols.

Authentication Trap
Enables or disables the reporting of SNMP authentication failures. If enabled, a notification trap is sent to the configured trap addresses in the event of an authentication failure. The default value is False.

BER
Bit Error Rate

Bit Error Rate
The probability that a transmitted bit will be erroneously received. The BER is measured by counting the number of bits in error at the output of a receiver and dividing by the total number of bits in the transmission. BER is typically expressed as a negative power of 10.

Buffer Credit
A measure of port buffer capacity equal to one frame.

Class 2 Service
A service that multiplexes frames at frame boundaries to or from one or more N_Ports wit h acknowledgment provided.

Class 3 Service
A service that multiplexes frames at frame boundaries to or from one or more N_Ports without acknowledgment.

Contact
Specifies the name of the contact person who is to be contacted to respond to trap events. The default is undefined.

Datagram
A message sent between two communicating entities for which no explicit link level acknowledgement is expected.

Domain ID
User defined name that identifies the switch in the fabric.

Fabric Management Switch
The switch through which the fabric is managed.

Flash Memory
Memory on the switch that contains the chassis control firmware.

Frame
Data unit consisting of a start-of-frame (SOF) delimiter, header, data payload, CRC, and an end-of-frame (EOF) delimiter.

ICMP
Internet Control Message Protocol

IETF
Internet Engineering Task Force

Initiator
The device that initiates a data exchange with a target device.

Internet Engineering Task Force
A large open international community of network designers, operators, vendors, and researchers concerned with evolution and smooth operation of the Internet, and responsible for producing RFCs. The standards body responsible for Internet standards, including SNMP, TCP/IP and policy for QoS.

Internet Control Message Protocol

A control protocol strongly related to IP and TCP, and used to convey a variety of control and error indications.

InteropCredit

Port configuration parameter that adjusts the number of port buffer credits to allow interoperability with some non-FC-SW2 compliant switches.

IP

Internet Protocol

ISLSecurity

ISLSecurity determines the switches that a port will establish a link with. Any—link with any switch. Ours—link only to another IBM Flex System FC3171 8 Gb SAN Switch and Pass-thru. None—the port will not establish an ISL link.

LCFEnable

LCFEnable gives preference to Link control frames (such as Class 2 ACK frames) over other frames, when queued for transmission in the switch. This may provide better performance when running Class 2 traffic. LCFEnable is incompatible with MFSEnable, and both cannot be selected. (True / False)

LIP

Loop Initialization Primitive sequence

Location

Specifies the switch location. The default is undefined.

Logged-In LED

A port LED that indicates device login or loop initialization status.

Management Information Base

A set of guidelines and definitions for the Fibre Channel functions.The specification and formal description of a set of objects and variables that can be read and possibly written using the SNMP protocol. Various standard MIBs are defined by the Internet Engineering Task Force.

Management Workstation

Workstation that manages the fabric through the fabric management switch.

MIB

Management Information Base

MSEnable

Determines whether GS-3 management server commands will be accepted on the port. It can be used to prevent in-band management of the switch on any or all ports. (True / False)

NL_Port

Node Loop Port. A Fibre Channel device port that supports arbitrated loop protocol.

N_Port

Node Port. A Fibre Channel device port in a point-to-point or fabric connection.

NMS

Network Management Station

Network Management Station

The console through which an administrator performs management functions.

NoClose

Causes the switch to keep the loop open, if no other device is arbitrating. It is intended to improve performance when there is a single L_Port device connected to the switch. (True / False).

Node

An addressable entity connected to an I/O bus or network. Used primarily to refer to computers, storage devices, and storage subsystems. The component of a node that connects to the bus or network is a port.

Object

In the context of access control, an entity to which access is controlled and/or usage is restricted to authorized subjects.

QoS

Quality of Service

POST

Power On Self Test

Power On Self Test (POST)

Diagnostics that the switch chassis performs at start up.

Private Device

A device that can communicate only with other devices on the same loop.

Private Loop

A loop of private devices connected to a single switch port.

Read Community

Read Community Authentication. A write-only field; the value on the switch and the SNMP management server must be the same. The default value is Public.

Request For Comment (RFC)

Internet-related specifications, including standards, experimental definitions, informational documents and best practice definitions, produced by the IETF.

Enterprise Fabric Suite

Switch management application.

SFF

Small Form-Factor transceiver.

SFP

Small Form-Factor Pluggable. A transceiver device, smaller than a GigaBit Interface Converter, that plugs into the Fibre Channel port.

Simple Network Management Protocol

The protocol governing network management and that allows monitoring of network devices.

SMI

Structure of Management Information

Small Form Factor

A transceiver device, smaller than a GigaBit Interface Converter, that is permanently attached to the circuit board.

Small Form-Factor Pluggable

A transceiver device, smaller than a GigaBit Interface Converter, that plugs into the Fibre Channel port.

SNMP

Simple Network Management Protocol

Structure of Management Information

A notation for setting or retrieving management variables over SNMP.

Target

A storage device that responds to an initiator device.

TCP

Transmission Control Protocol

Trap Address

Specifies the IP address to which SNMP traps are sent. The default is 127.0.0.1. A maximum of 5 trap addresses are supported.

Trap Community

Trap Community Authentication. A write-only field; the value on the switch and the SNMP management server must be the same. The default value is Public.

Trap Port

The port number on which the trap is set.

Trap Severity

Specifies a severity level to assign to the trap. Trap severity levels include Unknown, Emergency, Alert, Critical, Error, Warning, Notify, Info, Debug, and Mark

UDP

User Datagram Protocol

User Datagram Protocol

An Internet protocol that provides connection-less datagram delivery service to applications. Abbreviated UDP. UDP over Internet Protocol adds the ability to address multiple endpoints within a single network node to IP.

VIEnable

FC-VI. When enabled, VI preference frames will be transmitted ahead of other frames. (True / False)

Worldwide Name (WWN)

A unique 64-bit address assigned to a device by the device manufacturer.

Write Community

Write Community Authentication. A write-only field; the value on the switch and the SNMP management server must be the same. The default value is Private.

WWN

Worldwide Name

Zone

A set of ports or devices grouped together to control the exchange of information.

Zone Set

A set of zones grouped together. The active zone set defines the zoning for a fabric.

Index

A

Additional IP Objects 90
Additional TCP Objects 134
Address Translation Group 46
Agent 5
Alert 6, 405
atIfIndex 46
atNetAddress 48
atPhysAddress 47
atTable 46

C

Capability Group 491
Class 1 Accounting Table 469
Class 2 Accounting Table 479
Class 3 Accounting Table 486
Configuration Group 416
configurationChangeTime 217
Configuring switch 13
Connectivity Table 219
Connectivity Unit Group 214
connUnitConfigurationChangeTime 234
connUnitContact 241
connUnitControl 240
connUnitDomainId 229
connUnitEventCurrID 246
connUnitEventDescr 294
connUnitEventFilter 243
connUnitEventId 288
connUnitEventIndex 287
connUnitEventObject 293
connUnitEventSeverity 291
connUnitEventType 292
connUnitEventUnitId 286
connUnitGlobalId 220
connUnitId 219
connUnitInfo 239
connUnitLinkAgentAddressTypeY 304
connUnitLinkAgentAddressY 303
connUnitLinkAgentPortY 305
connUnitLinkConnIdY 307
connUnitLinkCurrIndex 308
connUnitLinkIndex 296
connUnitLinkNodeIdX 297
connUnitLinkNodeIdY 300
connUnitLinkPortNumber 301
connUnitLinkPortNumberX 298
connUnitLinkPortWwnX 299
connUnitLinkPortWwnY 302
connUnitLinkUnitId 295
connUnitLinkUnitTypeY 306
connUnitLocation 242
connUnitMaxEvents 245
connUnitModuleId 237
connUnitName 238
connUnitNumEvents 244
connUnitNumports 222
connUnitNumRevs 235
connUnitNumSensors 232
connUnitNumZones 236
connUnitPortControl 277
connUnitPortFCClassCap 265
connUnitPortFCClassOp 266
connUnitPortFCId 272
connUnitPortHWState 285
connUnitPortIndex 263
connUnitPortModuleType 270
connUnitPortName 279
connUnitPortNodeWwn 284
connUnitPortPhysicalNumber 280
connUnitPortProtocolCap 282
connUnitPortProtocolOp 283
connUnitPortRevision 274
connUnitPortSn 273
connUnitPortSpeed 276
connUnitPortStatCountAddressErrors 374
connUnitPortStatCountBBCreditZero 334
connUnitPortStatCountClass1FBSYFrames 342
connUnitPortStatCountClass1FRJTFrames 344
connUnitPortStatCountClass1PBSYFrames 343
connUnitPortStatCountClass1PRJTFrames 345
connUnitPortStatCountClass1RxFrames 340
connUnitPortStatCountClass1TxFrames 341
connUnitPortStatCountClass2FBSYFrames 348
connUnitPortStatCountClass2FRJTFrames 350
connUnitPortStatCountClass2PBSYFrames 349
connUnitPortStatCountClass2PRJTFrames 351
connUnitPortStatCountClass2RxFrames 346
connUnitPortStatCountClass2TxFrames 347
connUnitPortStatCountClass3Discards 354
connUnitPortStatCountClass3RxFrames 352
connUnitPortStatCountClass3TxFrames 353
connUnitPortStatCountDelimiterErrors 375

connUnitPortStatCountEncodingDisparityErrors 376
connUnitPortStatCountError 329
connUnitPortStatCountFBSYFrames 336
connUnitPortStatCountFramesTooLong 372
connUnitPortStatCountFramesTruncated 373
connUnitPortStatCountFRJTFrames 338
connUnitPortStatCountInputBuffersFull 335
connUnitPortStatCountInvalidCRC 366
connUnitPortStatCountInvalidOrderedSets 371
connUnitPortStatCountInvalidTxWords 367
connUnitPortStatCountLinkFailures 365
connUnitPortStatCountLossofSignal 369
connUnitPortStatCountLossofSynchronization 370
connUnitPortStatCountNumberLinkResets 361
connUnitPortStatCountNumberOfflineSequences 364
connUnitPortStatCountPBSYFrames 337
connUnitPortStatCountPrimitiveSequenceProtocolErrors 368
connUnitPortStatCountPRJTFrames 339
connUnitPortStatCountRxBroadcastObjects 357
connUnitPortStatCountRxElements 333
connUnitPortStatCountRxLinkResets 359
connUnitPortStatCountRxMulticastObjects 355
connUnitPortStatCountRxObjects 331
connUnitPortStatCountRxOfflineSequences 362
connUnitPortStatCountTxBroadcastObjects 358
connUnitPortStatCountTxElements 332
connUnitPortStatCountTxLinkResets 360
connUnitPortStatCountTxMulticastObjects 356
connUnitPortStatCountTxObjects 330
connUnitPortStatCountTxOfflineSequences 363
connUnitPortState1 267
connUnitPortStatIndex 328
connUnitPortStatObject 281
connUnitPortStatUnitId 327
connUnitPortStatus 268
connUnitPortTransmitterType 269
connUnitPortType 264
connUnitPortUnitId 262
connUnitPortWwn 271
connUnitPrincipal 231
connUnitProduct 225
connUnitProxyMaster 230
connUnitREventTime 289
connUnitRevsDescription 253
connUnitRevsIndex 251
connUnitRevsRevId 252
connUnitRevsUnitId 250
connUnitSensorCharacteristic 261
connUnitSensorIndex 255
connUnitSensorInfo 258
connUnitSensorMessage 259
connUnitSensorName 256
connUnitSensorStatus 257
connUnitSensorType 260
connUnitSensorUnitId 254
connUnitSEventTime 290
connUnitSn 226
connUnitSnsClassOfSvc 383
connUnitSnsFabricPortName 389
connUnitSnsFC4Type 386
connUnitSnsHardAddress 390
connUnitSnsId 378
connUnitSnsNodeIPAddress 384
connUnitSnsNodeName 382
connUnitSnsPortIdentifier 380
connUnitSnsPortIndex 379
connUnitSnsPortIPAddress 388
connUnitSnsPortName 381
connUnitSnsPortType 387
connUnitSnsProcAssoc 385
connUnitSnsSymbolicNodeName 392
connUnitSnsSymbolicPortName 391
connUnitState 223
connUnitStatus 224
connUnitStatusChangeTime 233
connUnitTableChangeTime 218
connUnitType 221
connUnitUpTime 227
connUnitUrl 228
Critical 6, 405

D

Debug 6, 405

E

EGP Group 142
EGP Neighbor Table 146
egpAs 161
egpInErrors 143
egpInMsgs 142
egpNeighAddr 147
egpNeighAs 148
egpNeighEventTrigger 160
egpNeighInErrMsgs 153
egpNeighInErrs 150
egpNeighInMsgs 149
egpNeighIntervalHello 157
egpNeighIntervalPoll 158
egpNeighMode 159
egpNeighOutErrMsgs 154
egpNeighOutErrs 152
egpNeighOutMsgs 151
egpNeighState 146
egpNeighStateDowns 156

egpNeighStateUps 155
egpNeighTable 146
egpOutErrors 145
egpOutMsgs 144
Emergency 6, 405
Enterprise Fabric Manager 13
Error 6, 405
Error Group 457
Event Table 286

F

fcFeElementName 417
fcFeFabricName 416
fcFeModuleCapacity 418
fcFeModuleDescr 419
fcFeModuleFxPortCapacity 423
fcFeModuleLastChange 422
fcFeModuleName 424
fcFeModuleObjectID 420
fcFeModuleOperStatus 421
fcFeModuleTable 419
fcFxPortAddressIdErrors 464
fcFxPortAdminMode 441
fcFxPortBbCredit 428
fcFxPortBbCreditAvailable 439
fcFxPortBbCreditModel 456
fcFxPortC1AccountingTable 469
fcFxPortC1ConnTime 478
fcFxPortC1Discards 473
fcFxPortC1FbsyFrames 474
fcFxPortC1FrjtFrames 475
fcFxPortC1InConnections 476
fcFxPortC1InFrames 469
fcFxPortC1InOctets 471
fcFxPortC1OutConnections 477
fcFxPortC1OutFrames 470
fcFxPortC1OutOctets 472
fcFxPortC2AccountingTable 479
fcFxPortC2Discards 483
fcFxPortC2FbsyFrames 484
fcFxPortC2FrjtFrames 485
fcFxPortC2InFrames 479
fcFxPortC2InOctets 481
fcFxPortC2OutFrames 480
fcFxPortC2OutOctets 482
fcFxPortC3Discards 490
fcFxPortC3InFrames 486
fcFxPortC3InOctets 488
fcFxPortC3OutFrames 487
fcFxPortC3OutOctets 489
fcFxPortCapBbCreditMax 493
fcFxPortCapBbCreditMin 494
fcFxPortCapClass2SeqDeliv 500
fcFxPortCapClass3SeqDeliv 501
fcFxPortCapCos 497
fcFxPortCapFcphVersionHigh 491
fcFxPortCapFcphVersionLow 492
fcFxPortCapHoldTimeMax 502
fcFxPortCapHoldTimeMin 503
fcFxPortCapIntermix 498
fcFxPortCapRxDataFieldSizeMax 495
fcFxPortCapRxDataFieldSizeMin 496
fcFxPortCapStackedConnMode 499
fcFxPortCapTable 491
fcFxPortClass2SeqDeliv 435
fcFxPortClass2SeqDelivAgreed 452
fcFxPortClass3SeqDeliv 436
fcFxPortClass3SeqDelivAgreed 453
fcFxPortConnectedNxPort 455
fcFxPortCosSuppAgreed 449
fcFxPortCosSupported 432
fcFxPortDelimiterErrors 463
fcFxPortEdtov 431
fcFxPortFcphVersionAgreed 446
fcFxPortFcphVersionHigh 426
fcFxPortFcphVersionLow 427
fcFxPortHoldTime 437
fcFxPortID 438
fcFxPortIntermixSuppAgreed 450
fcFxPortIntermixSupported 433
fcFxPortInvalidCrcs 462
fcFxPortInvalidTxWords 461
fcFxPortLinkFailures 457
fcFxPortLinkReseatIns 465
fcFxPortLinkResetOuts 466
fcFxPortName 425
fcFxPortNxPortBbCredit 447
fcFxPortNxPortName 454
fcFxPortNxPortRxDataFieldSize 448
fcFxPortOlsIns 467
fcFxPortOlsOuts 468
fcFxPortOperMode 440
fcFxPortPhysAdminStatus 442
FcFxPortPhysEntry 442
fcFxPortPhysLastChange 444
fcFxPortPhysOperStatus 443, 507
fcFxPortPhysRttov 445
fcFxPortPhysTable 442
fcFxPortPrimSeqProtoErrors 460
fcFxPortRatov 430
fcFxPortRxBufSize 429
fcFxPortSigLosses 459
fcFxPortStackedConnMode 434
fcFxPortStackedConnModeAgreed 451
fcFxPortSyncLosses 458
Fx Port Fabric Login Table 446

FxPort Configuration Table 425
FxPort Physical Level Table 442

G

Groups in MIB-II 15

I

ICMP Group 91
icmpInAddrMaskReps 103
icmpInAddrMasks 102
icmpInDestUnreachs 93
icmpInEchoReps 99
icmpInEchos 98
icmpInErrors 92
icmpInMsgs 91
icmpInParmProbs 95
icmpInRedirects 97
icmpInSrcQuenchs 96
icmpInTimeExcds 94
icmpInTimestampReps 101
icmpInTimestamps 100
icmpOutAddrMaskReps 116
icmpOutAddrMasks 115
icmpOutDestUnreachs 106
icmpOutEchoReps 112
icmpOutEchos 111
icmpOutErrors 105
icmpOutMsgs 104
icmpOutParmProbs 108
icmpOutRedirects 110
icmpOutSrcQuenchs 109
icmpOutTimeExcds 107
icmpOutTimestampReps 114
icmpOutTimestamps 113
ifAdminStatus 30
ifDescr 25
ifIndex 24
ifInDiscards 36
ifInErrors 37
ifInNUcastPkts 35
ifInOctets 33
ifInUcastPkts 34
ifInUnknownProtos 38
ifLastChange 32
ifMtu 27
ifNumber 23
ifOperStatus 31
ifOutDiscards 42
ifOutErrors 43
ifOutNUcastPkts 41
ifOutOctets 39
ifOutQLen 44
ifOutUcastPkts 40
ifPhysAddress 29
ifSpecific 45
ifSpeed 28
ifType 26
Info 6, 405
Interfaces Group 23
Interfaces Table 24
IP Address Table 68
IP Address Translation Table 86
IP Group 49
IP Routing Table 73
ipAddrTable 68
ipAdEntAddr 68
ipAdEntBcastAddr 71
ipAdEntIfIndex 69
ipAdEntNetMask 70
ipAdEntReasmMaxSize 72
ipDefaultTTL 50
ipForwarding 49
ipForwDatagrams 54
ipFragCreates 67
ipFragFails 66
ipFragOKs 65
ipInAddrErrors 53
ipInDelivers 57
ipInDiscards 56
ipInHdrErrors 52
ipInReceives 51
ipInUnknownProtos 55
ipNetToMediaIfIndex 86
ipNetToMediaNetAddress 88
ipNetToMediaPhysAddress 87
ipNetToMediaType 89
ipOutDiscards 59
ipOutNoRoutes 60
ipOutRequests 58
ipReasmFails 64
ipReasmOKs 63
ipReasmReqds 62
ipReasmTimeout 61
ipRouteAge 82
ipRouteDest 73
ipRouteIfIndex 74
ipRouteInfo 85
ipRouteMask 83
ipRouteMetric1 75
ipRouteMetric2 76
ipRouteMetric3 77
ipRouteMetric4 78
ipRouteMetric5 84
ipRouteNextHop 79

ipRouteProto 81
ipRouteTable 73
ipRouteType 80
ipRoutingDiscards 90

L

Link Table 295

M

Management information bases 15
Manager 5
Mark 405
MIB Definitions 415
MIB-II 15
Module Table 419

N

Notify 6, 405

P

Port Statistics Table 327
Port Table 262

Q

qlgcChFwDwldFileName 515
qlgcChFwDwldHostAddr 512
qlgcChFwDwldHostAddrType 511
qlgcChFwDwldHostPort 513
qlgcChFwDwldPathName 514
qlgcChFwOpRequest 510
qlgcChFwOpResult 509
qlgcChFwResetMethod 516
QuickTools 13

R

Revision Table 250
revisionNumber 213

S

Sensor Table 253
Simple Name Server Table 377
Simple Network Management Protocol 5
SNMP Group 171
snmpEnableAuthenTraps 198
snmpInASNParseErrs 176
snmpInBadCommunityNames 174
snmpInBadCommunityUses 175
snmpInBadValues 179
snmpInBadVersions 173
snmpInGenErrs 181
snmpInGetNexts 185
snmpInGetRequests 184
snmpInGetResponses 187
snmpInNoSuchNames 178
snmpInPkts 171
snmpInReadOnlys 180
snmpInSetRequests 186
snmpInTooBigs 177
snmpInTotalReqVars 182
snmpInTotalSetVars 183
snmpInTraps 188
snmpOutBadValues 191
snmpOutGenErrs 192
snmpOutGetNexts 194
snmpOutGetRequests 193
snmpOutGetResponses 196
snmpOutNoSuchNames 190
snmpOutPkts 172
snmpOutSetRequests 195
snmpOutTooBigs 189
snmpOutTraps 197
SNMPv1 5
SNMPv2c 5
Status Group 438
statusChangeTime 216
sysContact 19
sysDescr 16
sysLocation 21
sysName 20
sysObjectID 17
sysServices 22
System Group 16
systemURL 215
sysUpTime 18

T

TCP Connection Table 129
TCP Group 117
tcpActiveOpens 121

tcpAttemptFails 123
tcpConnLocalAddress 130
tcpConnLocalPort 131
tcpConnRemAddress 132
tcpConnRemPort 133
tcpConnState 129
tcpCurrEstab 125
tcpEstabResets 124
tcpInErrs 134
tcpInSegs 126
tcpMaxConn 120
tcpOutRsts 135
tcpOutSegs 127
tcpPassiveOpens 122
tcpRetransSegs 128
tcpRtoAlgorithm 117
tcpRtoMax 119
tcpRtoMin 118
Transmission Group 162
trap severity levels 6, 405
Trap Table 405
trapRegFilter 409
trapRegIpAddress 407
trapRegPort 408
trapRegRowState 410

U

UDP Group 136
UDP Listener Table 140
udpInDatagrams 136
udpInErrors 138
udpLocalAddress 140
udpLocalPort 141
udpNoPorts 137
udpOutDatagrams 139
Unknown 6, 405
uNumber 214

W

Warning 6, 405

Part Number: 88Y7940

Printed in USA

(1P) P/N: 88Y7940