FITELnet-F100

IPsec

MIB

FITELnet-F100 IPsec

FITELnet-F100　　　　　　　　　　MIB

1

FITELnet-F100 MIB

MIB

MIB	OID	SYNTAX	ACC
sysDescr	system 1	DisplayString	R
sysObjectID	system 2	ObjectID	R

MIB.........................MIB

OID.........................   ID

SYNTAX....................   MIB

ACC........................

                              R Read Only

                              R/W Read Write

                              R/W(R) RFC   R/W MIB   Read Only

                              R/C Read Create

                              R/C(R) RFC   R/C MIB   Read Only

                              TRAP

                              NTFY

                              - - -

MIB

OBJECT	sysDescr
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software.
OID	{ system 1 }
NOTE	"Vxx.xx [YYYY/MM/DD hh:mm] H' Vxx.xx , [YYYY/MM/DD hh:mm] , H ( ) FITELnet-F100 Version 01.17 [2004/12/24 18:15]

OBJECT . . . . MIB

SYNTAX . . . .   MIB

MAX-ACCESS . . . .   Read-write

Read-only

STATUS . . . .

DSCRPT . . . .

OID . . . .   ID

NOTE . . . .

2 MIB

2.1 RFC1213-MIB

MIB　　　MIB　　　　　　　　　　　　　　　MIB　 Obsoleted
　　MIB

```
mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }
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 }
egp OBJECT IDENTIFIER ::= { mib-2 8 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp OBJECT IDENTIFIER ::= { mib-2 11 }
```

MIB	OID	SYNTAX	ACC
sysDescr	system.1	OctetString	R
sysObjectID	system.2	ObjectID	R
sysUpTime	system.3	TimeTicks	R
sysContact	system.4	OctetString	R
sysName	system.5	OctetString	R
sysLocation	system.6	OctetString	R
sysServices	system.7	INTEGER	R
ifNumber	interfaces.1	INTEGER	R
ifTable	interfaces.2	Aggregate	--
ifEntry	ifTable.1	Aggregate	--
ifIndex	ifEntry.1	INTEGER	R
ifDescr	ifEntry.2	OctetString	R
ifType	ifEntry.3	IANAifType	R
ifMtu	ifEntry.4	INTEGER	R
ifSpeed	ifEntry.5	Gauge	R
ifPhysAddress	ifEntry.6	OctetString	R
ifAdminStatus	ifEntry.7	INTEGER	R
ifOperStatus	ifEntry.8	INTEGER	R
ifLastChange	ifEntry.9	TimeTicks	R
ifInOctets	ifEntry.10	Counter	R
ifInUcastPkts	ifEntry.11	Counter	R
ifInNUcastPkts	ifEntry.12	Counter	R
ifInDiscards	ifEntry.13	Counter	R

MIB	OID	SYNTAX	ACC
ifInErrors	ifEntry.14	Counter	R
ifInUnknownProtos	ifEntry.15	Counter	R
ifOutOctets	ifEntry.16	Counter	R
ifOutUcastPkts	ifEntry.17	Counter	R
ifOutNUcastPkts	ifEntry.18	Counter	R
ifOutDiscards	ifEntry.19	Counter	R
ifOutErrors	ifEntry.20	Counter	R
ifOutQLen	ifEntry.21	Gauge	R
ifSpecific	ifEntry.22	ObjectID	R
atTable	at.1	Aggregate	--
atEntry	atTable.1	Aggregate	--
atIfIndex	atEntry.1	INTEGER	R/W
atPhysAddress	atEntry.2	OctetString	R/W
atNetAddress	atEntry.3	NetworkAddress	R/W
ipForwarding	ip.1	INTEGER	R
ipDefaultTTL	ip.2	INTEGER	R
ipInReceives	ip.3	Counter	R
ipInHdrErrors	ip.4	Counter	R
ipInAddrErrors	ip.5	Counter	R
ipForwDatagrams	ip.6	Counter	R
ipInUnknownProtos	ip.7	Counter	R
ipInDiscards	ip.8	Counter	R
ipInDelivers	ip.9	Counter	R
ipOutRequests	ip.10	Counter	R
ipOutDiscards	ip.11	Counter	R
ipOutNoRoutes	ip.12	Counter	R
ipReasmTimeout	ip.13	INTEGER	R
ipReasmReqds	ip.14	Counter	R
ipReasmOKs	ip.15	Counter	R
ipReasmFails	ip.16	Counter	R
ipFragOKs	ip.17	Counter	R
ipFragFails	ip.18	Counter	R
ipFragCreates	ip.19	Counter	R
ipAddrTable	ip.20	Aggregate	--
ipAddrEntry	ipAddrTable.1	Aggregate	--
ipAdEntAddr	ipAddrEntry.1	IpAddress	R
ipAdEntIfIndex	ipAddrEntry.2	INTEGER	R
ipAdEntNetMask	ipAddrEntry.3	IpAddress	R
ipAdEntBcastAddr	ipAddrEntry.4	INTEGER	R
ipAdEntReasmMaxSize	ipAddrEntry.5	INTEGER	R
ipRouteTable	ip.21	Aggregate	--
ipRouteEntry	ipRouteTable.1	Aggregate	--
ipRouteDest	ipRouteEntry.1	IpAddress	R
ipRouteIfIndex	ipRouteEntry.2	INTEGER	R

MIB	OID	SYNTAX	ACC
ipRouteMetric1	ipRouteEntry.3	INTEGER	R
ipRouteMetric2	ipRouteEntry.4	INTEGER	R
ipRouteMetric3	ipRouteEntry.5	INTEGER	R
ipRouteMetric4	ipRouteEntry.6	INTEGER	R
ipRouteNextHop	ipRouteEntry.7	IpAddress	R
ipRouteType	ipRouteEntry.8	INTEGER	R
ipRouteProto	ipRouteEntry.9	INTEGER	R
ipRouteAge	ipRouteEntry.10	INTEGER	R
ipRouteMask	ipRouteEntry.11	IpAddress	R
ipRouteMetric5	ipRouteEntry.12	INTEGER	R
ipRouteInfo	ipRouteEntry.13	ObjectID	R
ipNetToMediaTable	ip.22	Aggregate	--
ipNetToMediaEntry	ipNetToMediaTable.1	Aggregate	--
ipNetToMediaIfIndex	ipNetToMediaEntry.1	INTEGER	R
ipNetToMediaPhysAddress	ipNetToMediaEntry.2	OctetString	R
ipNetToMediaNetAddress	ipNetToMediaEntry.3	IpAddress	R
ipNetToMediaType	ipNetToMediaEntry.4	INTEGER	R
ipRoutingDiscards	ip.23	Counter	R
icmpInMsgs	icmp.1	Counter	R
icmpInErrors	icmp.2	Counter	R
icmpInDestUnreachs	icmp.3	Counter	R
icmpInTimeExcds	icmp.4	Counter	R
icmpInParmProbs	icmp.5	Counter	R
icmpInSrcQuenchs	icmp.6	Counter	R
icmpInRedirects	icmp.7	Counter	R
icmpInEchos	icmp.8	Counter	R
icmpInEchoReps	icmp.9	Counter	R
icmpInTimestamps	icmp.10	Counter	R
icmpInTimestampReps	icmp.11	Counter	R
icmpInAddrMasks	icmp.12	Counter	R
icmpInAddrMaskReps	icmp.13	Counter	R
icmpOutMsgs	icmp.14	Counter	R
icmpOutErrors	icmp.15	Counter	R
icmpOutDestUnreachs	icmp.16	Counter	R
icmpOutTimeExcds	icmp.17	Counter	R
icmpOutParmProbs	icmp.18	Counter	R
icmpOutSrcQuenchs	icmp.19	Counter	R
icmpOutRedirects	icmp.20	Counter	R
icmpOutEchos	icmp.21	Counter	R
icmpOutEchoReps	icmp.22	Counter	R
icmpOutTimestamps	icmp.23	Counter	R
icmpOutTimestampReps	icmp.24	Counter	R
icmpOutAddrMasks	icmp.25	Counter	R
icmpOutAddrMaskReps	icmp.26	Counter	R

MIB	OID	SYNTAX	ACC
tcpRtoAlgorithm	tcp.1	INTEGER	R
tcpRtoMin	tcp.2	INTEGER	R
tcpRtoMax	tcp.3	INTEGER	R
tcpMaxConn	tcp.4	INTEGER	R
tcpActiveOpens	tcp.5	Counter	R
tcpPassiveOpens	tcp.6	Counter	R
tcpAttemptFails	tcp.7	Counter	R
tcpEstabResets	tcp.8	Counter	R
tcpCurrEstab	tcp.9	Gauge	R
tcpInSegs	tcp.10	Counter	R
tcpOutSegs	tcp.11	Counter	R
tcpRetransSegs	tcp.12	Counter	R
tcpConnTable	tcp.13	Aggregate	--
tcpConnEntry	tcpConnTable.1	Aggregate	--
tcpConnState	tcpConnEntry.1	INTEGER	R
tcpConnLocalAddress	tcpConnEntry.2	IpAddress	R
tcpConnLocalPort	tcpConnEntry.3	INTEGER	R
tcpConnRemAddress	tcpConnEntry.4	IpAddress	R
tcpConnRemPort	tcpConnEntry.5	INTEGER	R
tcpInErrs	tcp.14	Counter	R
tcpOutRsts	tcp.15	Counter	R
udpInDatagrams	udp.1	Counter	R
udpNoPorts	udp.2	Counter	R
udpInErrors	udp.3	Counter	R
udpOutDatagrams	udp.4	Counter	R
udpTable	udp.5	Aggregate	--
udpEntry	udpTable.1	Aggregate	--
udpLocalAddress	udpEntry.1	IpAddress	R
udpLocalPort	udpEntry.2	INTEGER	R
snmpInPkts	snmp.1	Counter	R
snmpOutPkts	snmp.2	Counter	R
snmpInBadVersions	snmp.3	Counter	R
snmpInBadCommunityNames	snmp.4	Counter	R
snmpInBadCommunityUses	snmp.5	Counter	R
snmpInASNParseErrs	snmp.6	Counter	R
snmpInTooBigs	snmp.8	Counter	R
snmpInNoSuchNames	snmp.9	Counter	R
snmpInBadValues	snmp.10	Counter	R
snmpInReadOnlys	snmp.11	Counter	R
snmpInGenErrs	snmp.12	Counter	R
snmpInTotalReqVars	snmp.13	Counter	R
snmpInTotalSetVars	snmp.14	Counter	R
snmpInGetRequests	snmp.15	Counter	R
snmpInGetNexts	snmp.16	Counter	R

MIB	OID	SYNTAX	ACC
snmpInSetRequests	snmp.17	Counter	R
snmpInGetResponses	snmp.18	Counter	R
snmpInTraps	snmp.19	Counter	R
snmpOutTooBigs	snmp.20	Counter	R
snmpOutNoSuchNames	snmp.21	Counter	R
snmpOutBadValues	snmp.22	Counter	R
snmpOutGenErrs	snmp.24	Counter	R
snmpOutGetRequests	snmp.25	Counter	R
snmpOutGetNexts	snmp.26	Counter	R
snmpOutSetRequests	snmp.27	Counter	R
snmpOutGetResponses	snmp.28	Counter	R
snmpOutTraps	snmp.29	Counter	R
snmpEnableAuthenTraps	snmp.30	INTEGER	R

2.2 RFC1215-MIB

MIB	OID	SYNTAX	ACC
coldStart	snmp.0.0		TRAP
linkDown	snmp.0.2 ifIndex		TRAP
linkUp	snmp.0.3 ifIndex		TRAP
authenticationFailure	snmp.0.4		TRAP

2.3 SNMPv2-MIB

snmpMIB OBJECT IDENTIFIER ::= { snmpModules 1 }
snmpMIBObjects OBJECT IDENTIFIER ::= { snmpMIB 1 }
system OBJECT IDENTIFIER ::= { mib-2 1 }
snmp OBJECT IDENTIFIER ::= { mib-2 11 }
snmpTrap OBJECT IDENTIFIER ::= { snmpMIBObjects 4 }
snmpTraps OBJECT IDENTIFIER ::= { snmpMIBObjects 5 }
snmpSet OBJECT IDENTIFIER ::= { snmpMIBObjects 6 }
snmpMIBConformance OBJECT IDENTIFIER ::= { snmpMIB 2 }
snmpMIBCompliances OBJECT IDENTIFIER ::= { snmpMIBConformance 1 }
snmpMIBGroups OBJECT IDENTIFIER ::= { snmpMIBConformance 2 }
snmpBasicCompliance OBJECT IDENTIFIER ::= { snmpMIBCompliances 2 }

MIB	OID	SYNTAX	ACC
coldStart	snmpTraps.1		TRAP
authenticationFailure	snmpTraps.5		TRAP
sysDescr	system 1	DisplayString	R
sysObjectID	system 2	ObjectID	R

MIB	OID	SYNTAX	ACC
sysUpTime	system.3	TimeTicks	R
sysContact	system.4	DisplayString	R
sysName	system.5	DisplayString	R
sysLocation	system.6	DisplayString	R
sysServices	system.7	INTEGER	R
snmpInPkts	snmp.1	Counter32	R
snmpInBadVersions	snmp.3	Counter32	R
snmpInBadCommunityNames	snmp.4	Counter32	R
snmpInBadCommunityUses	snmp.5	Counter32	R
snmpInASNParseErrs	snmp.6	Counter32	R
snmpEnableAuthenTraps	snmp.30	INTEGER	R
snmpTrapOID	snmpTrap.1	ObjectID	NTFY
snmpTrapEnterprise	snmpTrap.3	ObjectID	NTFY
snmpOutPkts	snmp.2	Counter32	R
snmpInTooBigs	snmp.8	Counter32	R
snmpInNoSuchNames	snmp.9	Counter32	R
snmpInBadValues	snmp.10	Counter32	R
snmpInReadOnlys	snmp.11	Counter32	R
snmpInGenErrs	snmp.12	Counter32	R
snmpInTotalReqVars	snmp.13	Counter32	R
snmpInTotalSetVars	snmp.14	Counter32	R
snmpInGetRequests	snmp.15	Counter32	R
snmpInGetNexts	snmp.16	Counter32	R
snmpInSetRequests	snmp.17	Counter32	R
snmpInGetResponses	snmp.18	Counter32	R
snmpInTraps	snmp.19	Counter32	R
snmpOutTooBigs	snmp.20	Counter32	R
snmpOutNoSuchNames	snmp.21	Counter32	R
snmpOutBadValues	snmp.22	Counter32	R
snmpOutGenErrs	snmp.24	Counter32	R
snmpOutGetRequests	snmp.25	Counter32	R
snmpOutGetNexts	snmp.26	Counter32	R
snmpOutSetRequests	snmp.27	Counter32	R
snmpOutGetResponses	snmp.28	Counter32	R
snmpOutTraps	snmp.29	Counter32	R

2.4 SNMPv2-SMI

```
org OBJECT IDENTIFIER ::= { iso 3 }
dod OBJECT IDENTIFIER ::= { org 6 }
internet OBJECT IDENTIFIER ::= { dod 1 }
directory OBJECT IDENTIFIER ::= { internet 1 }
mgmt OBJECT IDENTIFIER ::= { internet 2 }
mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
experimental OBJECT IDENTIFIER ::= { internet 3 }
private OBJECT IDENTIFIER ::= { internet 4 }
enterprises OBJECT IDENTIFIER ::= { private 1 }
security OBJECT IDENTIFIER ::= { internet 5 }
snmpV2 OBJECT IDENTIFIER ::= { internet 6 }
snmpDomains OBJECT IDENTIFIER ::= { snmpV2 1 }
snmpProxys OBJECT IDENTIFIER ::= { snmpV2 2 }
snmpModules OBJECT IDENTIFIER ::= { snmpV2 3 }
```

2.5 SNMPv2-TC

2.6 IP-MIB

```
ipMIB OBJECT IDENTIFIER ::= { mib-2 48 }
ip OBJECT IDENTIFIER ::= { mib-2 4 }
icmp OBJECT IDENTIFIER ::= { mib-2 5 }
ipMIBConformance OBJECT IDENTIFIER ::= { ipMIB 2 }
ipMIBCompliances OBJECT IDENTIFIER ::= { ipMIBConformance 1 }
ipMIBGroups OBJECT IDENTIFIER ::= { ipMIBConformance 2 }
ipMIBCompliance OBJECT IDENTIFIER ::= { ipMIBCompliances 1 }
```

MIB	OID	SYNTAX	ACC
ipForwarding	ip 1	INTEGER	R
ipDefaultTTL	ip 2	INTEGER	R
ipInReceives	ip 3	Counter32	R
ipInHdrErrors	ip 4	Counter32	R
ipInAddrErrors	ip 5	Counter32	R
ipForwDatagrams	ip 6	Counter32	R
ipInUnknownProtos	ip 7	Counter32	R
ipInDiscards	ip 8	Counter32	R
ipInDelivers	ip 9	Counter32	R
ipOutRequests	ip 10	Counter32	R
ipOutDiscards	ip 11	Counter32	R
ipOutNoRoutes	ip 12	Counter32	R

MIB	OID	SYNTAX	ACC
ipReasmTimeout	ip.13	Integer32	R
ipReasmReqds	ip.14	Counter32	R
ipReasmOKs	ip.15	Counter32	R
ipReasmFails	ip.16	Counter32	R
ipFragOKs	ip.17	Counter32	R
ipFragFails	ip.18	Counter32	R
ipFragCreates	ip.19	Counter32	R
ipAddrTable	ip.20	Aggregate	--
ipAddrEntry	ipAddrTable.1	Aggregate	--
ipAdEntAddr	ipAddrEntry.1	IpAddress	R
ipAdEntIfIndex	ipAddrEntry.2	INTEGER	R
ipAdEntNetMask	ipAddrEntry.3	IpAddress	R
ipAdEntBcastAddr	ipAddrEntry.4	INTEGER	R
ipAdEntReasmMaxSize	ipAddrEntry.5	INTEGER	R
ipNetToMediaTable	ip.22	Aggregate	--
ipNetToMediaEntry	ipNetToMediaTable.1	Aggregate	--
ipNetToMediaIfIndex	ipNetToMediaEntry.1	INTEGER	R
ipNetToMediaPhysAddress	ipNetToMediaEntry.2	PhysAddress	R
ipNetToMediaNetAddress	ipNetToMediaEntry.3	IpAddress	R
ipNetToMediaType	ipNetToMediaEntry.4	INTEGER	R
ipRoutingDiscards	ip.23	Counter32	R
icmpInMsgs	icmp.1	Counter32	R
icmpInErrors	icmp.2	Counter32	R
icmpInDestUnreachs	icmp.3	Counter32	R
icmpInTimeExcds	icmp.4	Counter32	R
icmpInParmProbs	icmp.5	Counter32	R
icmpInSrcQuenchs	icmp.6	Counter32	R
icmpInRedirects	icmp.7	Counter32	R
icmpInEchos	icmp.8	Counter32	R
icmpInEchoReps	icmp.9	Counter32	R
icmpInTimestamps	icmp.10	Counter32	R
icmpInTimestampReps	icmp.11	Counter32	R
icmpInAddrMasks	icmp.12	Counter32	R
icmpInAddrMaskReps	icmp.13	Counter32	R
icmpOutMsgs	icmp.14	Counter32	R
icmpOutErrors	icmp.15	Counter32	R
icmpOutDestUnreachs	icmp.16	Counter32	R
icmpOutTimeExcds	icmp.17	Counter32	R
icmpOutParmProbs	icmp.18	Counter32	R
icmpOutSrcQuenchs	icmp.19	Counter32	R
icmpOutRedirects	icmp.20	Counter32	R
icmpOutEchos	icmp.21	Counter32	R
icmpOutEchoReps	icmp.22	Counter32	R
icmpOutTimestamps	icmp.23	Counter32	R

MIB	OID	SYNTAX	ACC
icmpOutTimestampReps	icmp.24	Counter32	R
icmpOutAddrMasks	icmp.25	Counter32	R
icmpOutAddrMaskReps	icmp.26	Counter32	R

2.7 TCP-MIB

tcpMIB OBJECT IDENTIFIER ::= { mib-2 49 }
tcp OBJECT IDENTIFIER ::= { mib-2 6 }
tcpMIBConformance OBJECT IDENTIFIER ::= { tcpMIB 2 }
tcpMIBCompliances OBJECT IDENTIFIER ::= { tcpMIBConformance 1 }
tcpMIBGroups OBJECT IDENTIFIER ::= { tcpMIBConformance 2 }
tcpMIBCompliance OBJECT IDENTIFIER ::= { tcpMIBCompliances 1 }

MIB	OID	SYNTAX	ACC
tcpRtoAlgorithm	tcp.1	INTEGER	R
tcpRtoMin	tcp.2	Integer32	R
tcpRtoMax	tcp.3	Integer32	R
tcpMaxConn	tcp.4	Integer32	R
tcpActiveOpens	tcp.5	Counter32	R
tcpPassiveOpens	tcp.6	Counter32	R
tcpAttemptFails	tcp.7	Counter32	R
tcpEstabResets	tcp.8	Counter32	R
tcpCurrEstab	tcp.9	Gauge32	R
tcpInSegs	tcp.10	Counter32	R
tcpOutSegs	tcp.11	Counter32	R
tcpRetransSegs	tcp.12	Counter32	R
tcpConnTable	tcp.13	Aggregate	--
tcpConnEntry	tcpConnTable.1	Aggregate	--
tcpConnState	tcpConnEntry.1	INTEGER	R
tcpConnLocalAddress	tcpConnEntry.2	IpAddress	R
tcpConnLocalPort	tcpConnEntry.3	INTEGER	R
tcpConnRemAddress	tcpConnEntry.4	IpAddress	R
tcpConnRemPort	tcpConnEntry.5	INTEGER	R
tcpInErrs	tcp.14	Counter32	R
tcpOutRsts	tcp.15	Counter32	R

2.8 UDP-MIB

udpMIB OBJECT IDENTIFIER ::= { mib-2 50 }
udp OBJECT IDENTIFIER ::= { mib-2 7 }
udpMIBConformance OBJECT IDENTIFIER ::= { udpMIB 2 }
udpMIBCompliances OBJECT IDENTIFIER ::= { udpMIBConformance 1 }
udpMIBGroups OBJECT IDENTIFIER ::= { udpMIBConformance 2 }
udpMIBCompliance OBJECT IDENTIFIER ::= { udpMIBCompliances 1 }

MIB	OID	SYNTAX	ACC
udpInDatagrams	udp.1	Counter32	R
udpNoPorts	udp.2	Counter32	R
udpInErrors	udp.3	Counter32	R
udpOutDatagrams	udp.4	Counter32	R
udpTable	udp.5	Aggregate	--
udpEntry	udpTable.1	Aggregate	--
udpLocalAddress	udpEntry.1	IpAddress	R
udpLocalPort	udpEntry.2	INTEGER	R

2.9 IP-FORWARD-MIB

ipForward OBJECT IDENTIFIER ::= { ip 24 }
ipForwardConformance OBJECT IDENTIFIER ::= { ipForward 5 }
ipForwardGroups OBJECT IDENTIFIER ::= { ipForwardConformance 1 }
ipForwardCompliances OBJECT IDENTIFIER ::= { ipForwardConformance 2 }
ipForwardCompliance OBJECT IDENTIFIER ::= { ipForwardCompliances 1 }
ipForwardOldCompliance OBJECT IDENTIFIER ::= { ipForwardCompliances 2 }

MIB	OID	SYNTAX	ACC
ipForwardNumber	ipForward.1	Gauge32	R
ipForwardTable	ipForward.2	Aggregate	--
ipForwardEntry	ipForwardTable.1	Aggregate	--
ipForwardDest	ipForwardEntry.1	IpAddress	R
ipForwardMask	ipForwardEntry.2	IpAddress	R
ipForwardPolicy	ipForwardEntry.3	Integer32	R
ipForwardNextHop	ipForwardEntry.4	IpAddress	R
ipForwardIfIndex	ipForwardEntry.5	Integer32	R
ipForwardType	ipForwardEntry.6	INTEGER	R
ipForwardProto	ipForwardEntry.7	INTEGER	R
ipForwardAge	ipForwardEntry.8	Integer32	R
ipForwardInfo	ipForwardEntry.9	ObjectID	R
ipForwardNextHopAS	ipForwardEntry.10	Integer32	R
ipForwardMetric1	ipForwardEntry.11	Integer32	R
ipForwardMetric2	ipForwardEntry.12	Integer32	R
ipForwardMetric3	ipForwardEntry.13	Integer32	R

MIB	OID	SYNTAX	ACC
ipForwardMetric4	ipForwardEntry.14	Integer32	R
ipForwardMetric5	ipForwardEntry.15	Integer32	R

2.10 IF-MIB

ifMIB OBJECT IDENTIFIER ::= { mib-2 31 }
ifMIBObjects OBJECT IDENTIFIER ::= { ifMIB 1 }
interfaces OBJECT IDENTIFIER ::= { mib-2 2 }
ifConformance OBJECT IDENTIFIER ::= { ifMIB 2 }
ifGroups OBJECT IDENTIFIER ::= { ifConformance 1 }
ifCompliances OBJECT IDENTIFIER ::= { ifConformance 2 }
ifCompliance3 OBJECT IDENTIFIER ::= { ifCompliances 3 }
ifCompliance OBJECT IDENTIFIER ::= { ifCompliances 1 }
ifCompliance2 OBJECT IDENTIFIER ::= { ifCompliances 2 }

MIB	OID	SYNTAX	ACC
linkDown	snmpTraps.3 ifIndex, ifAdminStatus, ifOperStatus		TRAP
linkUp	snmpTraps.4 ifIndex, ifAdminStatus, ifOperStatus		TRAP
ifNumber	interfaces.1	Integer32	R
ifTableLastChange	ifMIBObjects.5	TimeTicks	R
ifTable	interfaces.2	Aggregate	--
ifEntry	ifTable.1	Aggregate	--
ifIndex	ifEntry.1	InterfaceIndex	R
ifDescr	ifEntry.2	DisplayString	R
ifType	ifEntry.3	IANAifType	R
ifMtu	ifEntry.4	Integer32	R
ifSpeed	ifEntry.5	Gauge32	R
ifPhysAddress	ifEntry.6	PhysAddress	R
ifAdminStatus	ifEntry.7	INTEGER	R
ifOperStatus	ifEntry.8	INTEGER	R
ifLastChange	ifEntry.9	TimeTicks	R
ifInOctets	ifEntry.10	Counter32	R
ifInUcastPkts	ifEntry.11	Counter32	R
ifInNUcastPkts	ifEntry.12	Counter32	R
ifInDiscards	ifEntry.13	Counter32	R
ifInErrors	ifEntry.14	Counter32	R
ifInUnknownProtos	ifEntry.15	Counter32	R
ifOutOctets	ifEntry.16	Counter32	R
ifOutUcastPkts	ifEntry.17	Counter32	R

MIB	OID	SYNTAX	ACC
ifOutNUcastPkts	ifEntry.18	Counter32	R
ifOutDiscards	ifEntry.19	Counter32	R
ifOutErrors	ifEntry.20	Counter32	R
ifOutQLen	ifEntry.21	Gauge32	R
ifSpecific	ifEntry.22	ObjectID	R
ifXTable	ifMIBObjects.1	Aggregate	--
ifXEntry	ifXTable.1	Aggregate	--
ifName	ifXEntry.1	DisplayString	R
ifInMulticastPkts	ifXEntry.2	Counter32	R
ifInBroadcastPkts	ifXEntry.3	Counter32	R
ifOutMulticastPkts	ifXEntry.4	Counter32	R
ifOutBroadcastPkts	ifXEntry.5	Counter32	R
ifHCInOctets	ifXEntry.6	Counter64	R
ifHCInUcastPkts	ifXEntry.7	Counter64	R
ifHCInMulticastPkts	ifXEntry.8	Counter64	R
ifHCInBroadcastPkts	ifXEntry.9	Counter64	R
ifHCOutOctets	ifXEntry.10	Counter64	R
ifHCOutUcastPkts	ifXEntry.11	Counter64	R
ifHCOutMulticastPkts	ifXEntry.12	Counter64	R
ifHCOutBroadcastPkts	ifXEntry.13	Counter64	R
ifLinkUpDownTrapEnable	ifXEntry.14	INTEGER	R
ifHighSpeed	ifXEntry.15	Gauge32	R
ifPromiscuousMode	ifXEntry.16	TruthValue	R
ifConnectorPresent	ifXEntry.17	TruthValue	R
ifAlias	ifXEntry.18	DisplayString	R
ifCounterDiscontinuityTime	ifXEntry.19	TimeStamp	R

2.11 IPV6-MIB

```
ipv6MIB OBJECT IDENTIFIER ::= { mib-2 55 }
ipv6MIBObjects OBJECT IDENTIFIER ::= { ipv6MIB 1 }
ipv6Notifications OBJECT IDENTIFIER ::= { ipv6MIB 2 }
ipv6NotificationPrefix OBJECT IDENTIFIER ::= { ipv6Notifications 0 }
ipv6Conformance OBJECT IDENTIFIER ::= { ipv6MIB 3 }
ipv6Compliances OBJECT IDENTIFIER ::= { ipv6Conformance 1 }
ipv6Groups OBJECT IDENTIFIER ::= { ipv6Conformance 2 }
ipv6Compliance OBJECT IDENTIFIER ::= { ipv6Compliances 1 }
```

MIB	OID	SYNTAX	ACC
ipv6Forwarding	ipv6MIBObjects.1	INTEGER	R
ipv6DefaultHopLimit	ipv6MIBObjects.2	INTEGER	R
ipv6Interfaces	ipv6MIBObjects.3	Unsigned32	R
ipv6IfTableLastChange	ipv6MIBObjects.4	TimeStamp	R
ipv6IfTable	ipv6MIBObjects.5	Aggregate	--

MIB	OID	SYNTAX	ACC
ipv6IfEntry	ipv6IfTable. 1	Aggregate	--
ipv6IfIndex	ipv6IfEntry. 1	Ipv6IfIndex	--
ipv6IfDescr	ipv6IfEntry. 2	DisplayString	R
ipv6IfLowerLayer	ipv6IfEntry. 3	VariablePointer	R
ipv6IfEffectiveMtu	ipv6IfEntry. 4	Unsigned32	R
ipv6IfReasmMaxSize	ipv6IfEntry. 5	Unsigned32	R
ipv6IfIdentifier	ipv6IfEntry. 6	Ipv6AddressIfIdentifier	R
ipv6IfIdentifierLength	ipv6IfEntry. 7	INTEGER	R
ipv6IfPhysicalAddress	ipv6IfEntry. 8	PhysAddress	R
ipv6IfAdminStatus	ipv6IfEntry. 9	INTEGER	R
ipv6IfOperStatus	ipv6IfEntry. 10	INTEGER	R
ipv6IfLastChange	ipv6IfEntry. 11	TimeStamp	R
ipv6IfStatsTable	ipv6MIBObjects. 6	Aggregate	--
ipv6IfStatsEntry	ipv6IfStatsTable. 1	Aggregate	--
ipv6IfStatsInReceives	ipv6IfStatsEntry. 1	Counter32	R
ipv6IfStatsInHdrErrors	ipv6IfStatsEntry. 2	Counter32	R
ipv6IfStatsInTooBigErrors	ipv6IfStatsEntry. 3	Counter32	R
ipv6IfStatsInNoRoutes	ipv6IfStatsEntry. 4	Counter32	R
ipv6IfStatsInAddrErrors	ipv6IfStatsEntry. 5	Counter32	R
ipv6IfStatsInUnknownProtos	ipv6IfStatsEntry. 6	Counter32	R
ipv6IfStatsInTruncatedPkts	ipv6IfStatsEntry. 7	Counter32	R
ipv6IfStatsInDiscards	ipv6IfStatsEntry. 8	Counter32	R
ipv6IfStatsInDelivers	ipv6IfStatsEntry. 9	Counter32	R
ipv6IfStatsOutForwDatagrams	ipv6IfStatsEntry. 10	Counter32	R
ipv6IfStatsOutRequests	ipv6IfStatsEntry. 11	Counter32	R
ipv6IfStatsOutDiscards	ipv6IfStatsEntry. 12	Counter32	R
ipv6IfStatsOutFragOKs	ipv6IfStatsEntry. 13	Counter32	R
ipv6IfStatsOutFragFails	ipv6IfStatsEntry. 14	Counter32	R
ipv6IfStatsOutFragCreates	ipv6IfStatsEntry. 15	Counter32	R
ipv6IfStatsReasmReqds	ipv6IfStatsEntry. 16	Counter32	R
ipv6IfStatsReasmOKs	ipv6IfStatsEntry. 17	Counter32	R
ipv6IfStatsReasmFails	ipv6IfStatsEntry. 18	Counter32	R
ipv6IfStatsInMcastPkts	ipv6IfStatsEntry. 19	Counter32	R
ipv6IfStatsOutMcastPkts	ipv6IfStatsEntry. 20	Counter32	R
ipv6AddrTable	ipv6MIBObjects. 8	Aggregate	--
ipv6AddrEntry	ipv6AddrTable. 1	Aggregate	--
ipv6AddrAddress	ipv6AddrEntry. 1	Ipv6Address	--
ipv6AddrPfxLength	ipv6AddrEntry. 2	INTEGER	R
ipv6AddrType	ipv6AddrEntry. 3	INTEGER	R
ipv6AddrAnycastFlag	ipv6AddrEntry. 4	TruthValue	R
ipv6AddrStatus	ipv6AddrEntry. 5	INTEGER	R
ipv6RouteNumber	ipv6MIBObjects. 9	Gauge32	R
ipv6DiscardedRoutes	ipv6MIBObjects. 10	Counter32	R
ipv6RouteTable	ipv6MIBObjects. 11	Aggregate	--

MIB	OID	SYNTAX	ACC
ipv6RouteEntry	ipv6RouteTable.1	Aggregate	--
ipv6RouteDest	ipv6RouteEntry.1	Ipv6Address	--
ipv6RoutePfxLength	ipv6RouteEntry.2	INTEGER	--
ipv6RouteIndex	ipv6RouteEntry.3	Unsigned32	--
ipv6RouteIfIndex	ipv6RouteEntry.4	Ipv6IfIndexOrZero	R
ipv6RouteNextHop	ipv6RouteEntry.5	Ipv6Address	R
ipv6RouteType	ipv6RouteEntry.6	INTEGER	R
ipv6RouteProtocol	ipv6RouteEntry.7	INTEGER	R
ipv6RoutePolicy	ipv6RouteEntry.8	Integer32	R
ipv6RouteAge	ipv6RouteEntry.9	Unsigned32	R
ipv6RouteNextHopRDI	ipv6RouteEntry.10	Unsigned32	R
ipv6RouteMetric	ipv6RouteEntry.11	Unsigned32	R
ipv6RouteWeight	ipv6RouteEntry.12	Unsigned32	R
ipv6RouteInfo	ipv6RouteEntry.13	RowPointer	R
ipv6RouteValid	ipv6RouteEntry.14	TruthValue	R

2.12 IPV6-ICMP-MIB

ipv6IcmpMIB OBJECT IDENTIFIER ::= { mib-2 56 }
ipv6IcmpMIBObjects OBJECT IDENTIFIER ::= { ipv6IcmpMIB 1 }
ipv6IcmpConformance OBJECT IDENTIFIER ::= { ipv6IcmpMIB 2 }
ipv6IcmpCompliances OBJECT IDENTIFIER ::= { ipv6IcmpConformance 1 }
ipv6IcmpGroups OBJECT IDENTIFIER ::= { ipv6IcmpConformance 2 }
ipv6IcmpCompliance OBJECT IDENTIFIER ::= { ipv6IcmpCompliances 1 }

MIB	OID	SYNTAX	ACC
ipv6IfIcmpTable	ipv6IcmpMIBObjects.1	Aggregate	--
ipv6IfIcmpEntry	ipv6IfIcmpTable.1	Aggregate	--
ipv6IfIcmpInMsgs	ipv6IfIcmpEntry.1	Counter32	R
ipv6IfIcmpInErrors	ipv6IfIcmpEntry.2	Counter32	R
ipv6IfIcmpInDestUnreachs	ipv6IfIcmpEntry.3	Counter32	R
ipv6IfIcmpInAdminProhibs	ipv6IfIcmpEntry.4	Counter32	R
ipv6IfIcmpInTimeExcds	ipv6IfIcmpEntry.5	Counter32	R
ipv6IfIcmpInParmProblems	ipv6IfIcmpEntry.6	Counter32	R
ipv6IfIcmpInPktTooBigs	ipv6IfIcmpEntry.7	Counter32	R
ipv6IfIcmpInEchos	ipv6IfIcmpEntry.8	Counter32	R
ipv6IfIcmpInEchoReplies	ipv6IfIcmpEntry.9	Counter32	R
ipv6IfIcmpInRouterSolicits	ipv6IfIcmpEntry.10	Counter32	R
ipv6IfIcmpInRouterAdvertisements	ipv6IfIcmpEntry.11	Counter32	R
ipv6IfIcmpInNeighborSolicits	ipv6IfIcmpEntry.12	Counter32	R
ipv6IfIcmpInNeighborAdvertisements	ipv6IfIcmpEntry.13	Counter32	R
ipv6IfIcmpInRedirects	ipv6IfIcmpEntry.14	Counter32	R
ipv6IfIcmpInGroupMembQueries	ipv6IfIcmpEntry.15	Counter32	R

MIB	OID	SYNTAX	ACC
ipv6IfIcmpInGroupMembResponses	ipv6IfIcmpEntry.16	Counter32	R
ipv6IfIcmpInGroupMembReductions	ipv6IfIcmpEntry.17	Counter32	R
ipv6IfIcmpOutMsgs	ipv6IfIcmpEntry.18	Counter32	R
ipv6IfIcmpOutErrors	ipv6IfIcmpEntry.19	Counter32	R
ipv6IfIcmpOutDestUnreachs	ipv6IfIcmpEntry.20	Counter32	R
ipv6IfIcmpOutAdminProhibs	ipv6IfIcmpEntry.21	Counter32	R
ipv6IfIcmpOutTimeExcds	ipv6IfIcmpEntry.22	Counter32	R
ipv6IfIcmpOutParmProblems	ipv6IfIcmpEntry.23	Counter32	R
ipv6IfIcmpOutPktTooBigs	ipv6IfIcmpEntry.24	Counter32	R
ipv6IfIcmpOutEchos	ipv6IfIcmpEntry.25	Counter32	R
ipv6IfIcmpOutEchoReplies	ipv6IfIcmpEntry.26	Counter32	R
ipv6IfIcmpOutRouterSolicits	ipv6IfIcmpEntry.27	Counter32	R
ipv6IfIcmpOutRouterAdvertisements	ipv6IfIcmpEntry.28	Counter32	R
ipv6IfIcmpOutNeighborSolicits	ipv6IfIcmpEntry.29	Counter32	R
ipv6IfIcmpOutNeighborAdvertisements	ipv6IfIcmpEntry.30	Counter32	R
ipv6IfIcmpOutRedirects	ipv6IfIcmpEntry.31	Counter32	R
ipv6IfIcmpOutGroupMembQueries	ipv6IfIcmpEntry.32	Counter32	R
ipv6IfIcmpOutGroupMembResponses	ipv6IfIcmpEntry.33	Counter32	R
ipv6IfIcmpOutGroupMembReductions	ipv6IfIcmpEntry.34	Counter32	R

2.13 IANAifType-MIB

ianaifType OBJECT IDENTIFIER ::= { mib-2 30 }

2.14 INET-ADDRESS-MIB

inetAddressMIB OBJECT IDENTIFIER ::= { mib-2 76 }

2.15 FURUKAWA-MIB

furukawa OBJECT IDENTIFIER ::= { enterprises 246 }
products OBJECT IDENTIFIER ::= { furukawa 1 }
temporary OBJECT IDENTIFIER ::= { furukawa 2 }
furukawaAdmin OBJECT IDENTIFIER ::= { furukawa 3 }
infonet OBJECT IDENTIFIER ::= { products 1 }
infonetBase OBJECT IDENTIFIER ::= { infonet 1 }
infonetMgt OBJECT IDENTIFIER ::= { infonet 29 }

2.16 FURUKAWA-INFONETBASE-MIB

infonetBase OBJECT IDENTIFIER ::= { infonet 1 }
infonetSystem OBJECT IDENTIFIER ::= { infonetBase 1 }
infonetSystemError OBJECT IDENTIFIER ::= { infonetBase 2 }
infonetSystemLineError OBJECT IDENTIFIER ::= { infonetBase 3 }
infonetMbufUsage OBJECT IDENTIFIER ::= { infonetBase 4 }
infonetEvent OBJECT IDENTIFIER ::= { infonetBase 5 }
infonetCpuBusy OBJECT IDENTIFIER ::= { infonetBase 6 }

MIB	OID	SYNTAX	ACC
infSystemErrorPoint	infonetSystemError.1	OctetString	R
infSystemErrorText1	infonetSystemError.2	DisplayString	R
infSystemErrorText2	infonetSystemError.3	DisplayString	R
infSystemErrorText3	infonetSystemError.4	DisplayString	R
infSystemErrorText4	infonetSystemError.5	DisplayString	R
infSystemErrorText5	infonetSystemError.6	DisplayString	R
infSystemErrorText6	infonetSystemError.7	DisplayString	R
infSystemErrorText7	infonetSystemError.8	DisplayString	R
infSystemErrorText8	infonetSystemError.9	DisplayString	R
infSystemErrorText9	infonetSystemError.10	DisplayString	R
infSystemErrorText10	infonetSystemError.11	DisplayString	R
infSystemErrorText11	infonetSystemError.12	DisplayString	R
infSystemErrorText12	infonetSystemError.13	DisplayString	R
infSystemErrorText13	infonetSystemError.14	DisplayString	R
infSystemErrorText14	infonetSystemError.15	DisplayString	R
infSystemErrorText15	infonetSystemError.16	DisplayString	R
infSystemErrorText16	infonetSystemError.17	DisplayString	R
infSystemErrorText17	infonetSystemError.18	DisplayString	R
infSystemErrorText18	infonetSystemError.19	DisplayString	R
infSystemErrorText19	infonetSystemError.20	DisplayString	R
infSystemErrorText20	infonetSystemError.21	DisplayString	R

2.17 INFONET-PRODUCTID-MIB

```
infonetProductID OBJECT IDENTIFIER ::= { infonet 18 }
infProdRouter OBJECT IDENTIFIER ::= { infonetProductID 1 }
infProdFW OBJECT IDENTIFIER ::= { infonetProductID 2 }
infProdAJ OBJECT IDENTIFIER ::= { infonetProductID 3 }
infProdGX20 OBJECT IDENTIFIER ::= { infProdRouter 1 }
infProdGX21 OBJECT IDENTIFIER ::= { infProdRouter 2 }
infProdGX30 OBJECT IDENTIFIER ::= { infProdRouter 3 }
infProdGX31 OBJECT IDENTIFIER ::= { infProdRouter 4 }
infProdG10 OBJECT IDENTIFIER ::= { infProdRouter 5 }
infProdG40 OBJECT IDENTIFIER ::= { infProdRouter 6 }
infProdAX160 OBJECT IDENTIFIER ::= { infProdRouter 7 }
infProdG12 OBJECT IDENTIFIER ::= { infProdRouter 8 }
infProdB10 OBJECT IDENTIFIER ::= { infProdRouter 9 }
infProdB8 OBJECT IDENTIFIER ::= { infProdRouter 10 }
infProdR10 OBJECT IDENTIFIER ::= { infProdRouter 11 }
infProdF40 OBJECT IDENTIFIER ::= { infProdRouter 12 }
infProdG20 OBJECT IDENTIFIER ::= { infProdRouter 13 }
infProdG80 OBJECT IDENTIFIER ::= { infProdRouter 14 }
infProdR20 OBJECT IDENTIFIER ::= { infProdRouter 15 }
infProdF100 OBJECT IDENTIFIER ::= { infProdRouter 16 }
infProdG21 OBJECT IDENTIFIER ::= { infProdRouter 17
infProdG1000 OBJECT IDENTIFIER ::= { infProdRouter 18 }
infProdFM4 OBJECT IDENTIFIER ::= { infProdFW 1 }
infProdFM6 OBJECT IDENTIFIER ::= { infProdFW 2 }
infProdFMRmaster OBJECT IDENTIFIER ::= { infProdFW 3 }
infProdFMRslave OBJECT IDENTIFIER ::= { infProdFW 4 }
infProdFMLight OBJECT IDENTIFIER ::= { infProdFW 5 }
infProdAJ10 OBJECT IDENTIFIER ::= { infProdAJ 1 }
```

2.18 FURUKAWA-INFMGTPROCESS-MIB

```
infMgtProcs OBJECT IDENTIFIER ::= { infonetMgt 6 }
infMgtProcTotalCpu OBJECT IDENTIFIER ::= { infMgtProcs 3 }
infMgtProcTotalMem OBJECT IDENTIFIER ::= { infMgtProcs 4 }
infMgtProcConformance OBJECT IDENTIFIER ::= { infMgtProcs 2 }
infMgtProcCompliances OBJECT IDENTIFIER ::= { infMgtProcConformance 1 }
infMgtProcGroups OBJECT IDENTIFIER ::= { infMgtProcConformance 2 }
infMgrProcCompliance OBJECT IDENTIFIER ::= { infMgtProcCompliances 1 }
```

MIB	OID	SYNTAX	ACC
infMgtProcTotalCPU5sec	infMgtProcTotalCpu 1	Gauge32	R
infMgtProcTotalCPU1min	infMgtProcTotalCpu 2	Gauge32	R
infMgtProcTotalCPU5min	infMgtProcTotalCpu 3	Gauge32	R

2.19   FURUKAWA-INFMGT-PATHCHECK-MIB

infMgtPcheck OBJECT IDENTIFIER ::= { infonetMgt 14 }
infMgtPcheckLst OBJECT IDENTIFIER ::= { infMgtPcheck 1 }
infMgtPcheckFil OBJECT IDENTIFIER ::= { infMgtPcheck 2 }
infMgtPcheckConf OBJECT IDENTIFIER ::= { infMgtPcheck 3 }
infMgtPcheckConfCompliances OBJECT IDENTIFIER ::= { infMgtPcheckConf 1 }
infMgtPcheckConfGroups OBJECT IDENTIFIER ::= { infMgtPcheckConf 2 }
infMgtPcheckConfCompliance OBJECT IDENTIFIER ::= { infMgtPcheckConfCompliances 1 }

MIB	OID	SYNTAX	ACC
infMgtPcheckLstNum	infMgtPcheckLst.1	Integer32	R
infMgtPcheckLstTable	infMgtPcheckLst.2	Aggregate	--
infMgtPcheckLstEntry	infMgtPcheckLstTable.1	Aggregate	--
infMgtPcheckLstIndex	infMgtPcheckLstEntry.1	Integer32	--
infMgtPcheckLstDstAddrType	infMgtPcheckLstEntry.2	InetAddressType	R
infMgtPcheckLstDstAddr	infMgtPcheckLstEntry.3	InetAddress	R
infMgtPcheckLstNormalCond	infMgtPcheckLstEntry.4	TruthValue	R
infMgtPcheckLstCurrentFails	infMgtPcheckLstEntry.5	Gauge32	R
infMgtPcheckLstCurrentSuccess	infMgtPcheckLstEntry.6	Gauge32	R
infMgtPcheckLstFailCond	infMgtPcheckLstEntry.7	Integer32	R
infMgtPcheckLstSuccessCond	infMgtPcheckLstEntry.8	Integer32	R
infMgtPcheckLstTrials	infMgtPcheckLstEntry.9	Integer32	R
infMgtPcheckLstInterval	infMgtPcheckLstEntry.10	Integer32	R
infMgtPcheckLstReducedInterval	infMgtPcheckLstEntry.11	Integer32	R
infMgtPcheckLstLastChange	infMgtPcheckLstEntry.12	TimeStamp	R
infMgtPcheckFilNum	infMgtPcheckFil.1	Integer32	R
infMgtPcheckFilTable	infMgtPcheckFil.2	Aggregate	--
infMgtPcheckFilEntry	infMgtPcheckFilTable.1	Aggregate	--
infMgtPcheckFilIndex	infMgtPcheckFilEntry.1	Integer32	--
infMgtPcheckFilPListIndex	infMgtPcheckFilEntry.2	Integer32	R
infMgtPcheckFilDstAddrType	infMgtPcheckFilEntry.3	InetAddressType	R
infMgtPcheckFilDstAddr	infMgtPcheckFilEntry.4	InetAddress	R
infMgtPcheckFilDstPfxLen	infMgtPcheckFilEntry.5	InetAddressPrefixLength	R
infMgtPcheckFilNextIfIndex1	infMgtPcheckFilEntry.6	InterfaceIndex	R
infMgtPcheckFilNextIfName1	infMgtPcheckFilEntry.7	DisplayString	R
infMgtPcheckFilNextIfIndex2	infMgtPcheckFilEntry.8	InterfaceIndexOrZero	R
infMgtPcheckFilNextIfName2	infMgtPcheckFilEntry.9	DisplayString	R

2.20 FURUKAWA-INFMGT-SYSTEM-RESOURCES-MIB

infMgtSystemResourcesMIB OBJECT IDENTIFIER ::= { infonetMgt 17 }
infMgtSysResrcMIBObjects OBJECT IDENTIFIER ::= { infMgtSystemResourcesMIB 1 }
infMgtSysResrcMIBNotificationPrefix OBJECT IDENTIFIER::={ infMgtSystemResourcesMIB 2 }
infMgtSysResrcMIBConformance OBJECT IDENTIFIER ::= { infMgtSystemResourcesMIB 3 }
imSysResrcNotifications OBJECT IDENTIFIER::={ infMgtSysResrcMIBNotificationPrefix 0 }
imSysResrcNotificationValues OBJECT IDENTIFIER::={ infMgtSysResrcMIBNotificationPrefix 1 }
infMgtSysResrcMIBGroups OBJECT IDENTIFIER ::= { infMgtSysResrcMIBConformance 1 }
infMgtSysResrcMIBCompliances OBJECT IDENTIFIER ::= { infMgtSysResrcMIBConformance 2 }
infMgtSysResrcMIBCompliance OBJECT IDENTIFIER ::= { infMgtSysResrcMIBCompliances 1 }

MIB	OID	SYNTAX	ACC
imSysResrcTable	infMgtSysResrcMIBObjects. 1	Aggregate	--
imSysResrcEntry	imSysResrcTable. 1	Aggregate	--
imSysResrcIndex	imSysResrcEntry. 1	InfMgtSysResrcIndex	--
imSysResrcName	imSysResrcEntry. 2	InfMgtSysResrcString	R
imSysResrcMaxEntries	imSysResrcEntry. 3	Integer32	R
imSysResrcReservedEntries	imSysResrcEntry. 4	Integer32	R
imSysResrcActiveEntries	imSysResrcEntry. 5	InfMgtSysResrcGauge	R
imSysResrcPeakEntries	imSysResrcEntry. 6	InfMgtSysResrcGauge	R
imSysResrcLimitFails	imSysResrcEntry. 7	InfMgtSysResrcCounter	R

3 MIB

3.1 RFC1213-MIB

MIB
MIB

3.2 RFC1213-MIB

OBJECT	atTable
SYNTAX	SEQUENCE OF AtEntry
ACCESS	not-accessible
STATUS	deprecated
DSCRPT	The Address Translation tables contain the NetworkAddress to `physical' address equivalences. Some interfaces do not use translation tables for determining address equivalences (e.g., DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation table is empty, i.e., has zero entries.
OID	{ at 1 }
NOTE	

OBJECT	atEntry
SYNTAX	AtEntry
ACCESS	not-accessible
STATUS	deprecated
DSCRPT	Each entry contains one NetworkAddress to `physical' address equivalence.
INDEX	atIfIndex, atNetAddress
OID	{ atTable 1 }
NOTE	

OBJECT	atIfIndex
SYNTAX	INTEGER
ACCESS	read-write
STATUS	deprecated
DSCRPT	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.

OID	{ atEntry 1 }
NOTE	

OBJECT	atPhysAddress
SYNTAX	PhysAddress
ACCESS	read-write
STATUS	deprecated
DSCRPT	The media-dependent `physical' address. Setting this object to a null string (one of zero length) has the effect of invaliding the corresponding entry in the atTable object. That is, it effectively dissasociates the interface identified with said entry from the mapping identified with said 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 atPhysAddress object.
OID	{ atEntry 2 }
NOTE	

OBJECT	atNetAddress
SYNTAX	NetworkAddress
ACCESS	read-write
STATUS	deprecated
DSCRPT	The NetworkAddress (e.g., the IP address) corresponding to the media-dependent `physical' address.
OID	{ atEntry 3 }
NOTE	

OBJECT	ipRouteTable
SYNTAX	SEQUENCE OF IpRouteEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	This entity's IP Routing table.
OID	{ ip 21 }
NOTE	

OBJECT	ipRouteEntry
SYNTAX	IpRouteEntry

ACCESS	not-accessible
STATUS	mandatory
DSCRPT	A route to a particular destination.
INDEX	ipRouteDest
OID	{ ipRouteTable 1 }
NOTE	

OBJECT	ipRouteDest
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 1 }
NOTE	

OBJECT	ipRouteIfIndex
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	The index value which 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.
OID	{ ipRouteEntry 2 }
NOTE	

OBJECT	ipRouteMetric1
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 3 }
NOTE	

OBJECT	ipRouteMetric2
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 4 }
NOTE	

OBJECT	ipRouteMetric3
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 5 }
NOTE	

OBJECT	ipRouteMetric4
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 6 }
NOTE	

OBJECT	ipRouteNextHop
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	The IP address of the next hop of this route. (In the case of a route bound to an interface which is realized via a broadcast media, the value of this field is the agent's IP address on that

	interface.)
OID	{ ipRouteEntry 7 }
NOTE	

OBJECT	ipRouteType
SYNTAX	INTEGER { other(1), invalid(2), direct(3), indirect(4) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	The type of route. Note that 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 dissasociates the destination identified with said entry from the route identified with said 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.
OID	{ ipRouteEntry 8 }
NOTE	

OBJECT	ipRouteProto
SYNTAX	INTEGER { other(1), local(2), netmgmt(3), icmp(4), egp(5), ggp(6), hello(7), rip(8), is-is(9), es-is(10), ciscoIgrp(11), bbnSpfIgp(12), ospf(13), bgp(14) }
ACCESS	read-only
STATUS	mandatory
DSCRPT	The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.
OID	{ ipRouteEntry 9 }
NOTE	

OBJECT	ipRouteAge
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory

DSCRPT	The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of `too old' can be implied except through knowledge of the routing protocol by which the route was learned.
OID	{ ipRouteEntry 10 }
NOTE	

OBJECT	ipRouteMask
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using one of: mask network 255.0.0.0 class-A 255.255.0.0 class-B 255.255.255.0 class-C If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism.
OID	{ ipRouteEntry 11 }
NOTE	

OBJECT	ipRouteMetric5
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	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.
OID	{ ipRouteEntry 12 }
NOTE	

OBJECT	ipRouteInfo
SYNTAX	OBJECT IDENTIFIER

ACCESS	read-only
STATUS	mandatory
DSCRPT	A reference to MIB definitions specific to the particular routing protocol which 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 syntatically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.
OID	{ ipRouteEntry 13 }
NOTE	

3.3 SNMPv2-MIB

MODULE	snmpMIB
UPDATED	9511090000Z
DSCRPT	The MIB module for SNMPv2 entities.
REVISION	9304010000Z
DSCRPT	The initial revision of this MIB module was published as RFC 1450.
OID	{ snmpModules 1 }
NOTE	

OBJECT	sysDescr
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software.
OID	{ system 1 }
NOTE	

OBJECT	sysObjectID
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises 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'.
OID	{ system 2 }
NOTE	

OBJECT	sysUpTime
SYNTAX	TimeTicks
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The time (in hundredths of a second) since the network management portion of the system was last re-initialized.
OID	{ system 3 }
NOTE	

OBJECT	sysContact
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The textual identification of the contact person for this managed node, together with information on how to contact this person. If no contact information is known, the value is the zero-length string.
OID	{ system 4 }
NOTE	

OBJECT	sysName
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-write
STATUS	current
DSCRPT	An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string.
OID	{ system 5 }
NOTE	

OBJECT	sysLocation
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The physical location of this node (e.g., `telephone

	closet, 3rd floor'). If the location is unknown, the value is the zero-length string.
OID	{ system 6 }
NOTE	

OBJECT	sysServices
SYNTAX	INTEGER (0..127)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A value which indicates the set of services that this entity may potentially 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 which performs only routing functions would have a value of 4 (2^(3-1)). In contrast, a node which is a host offering application services would have a value of 72 (2^(4-1) + 2^(7-1)). Note that in the context of the Internet suite of protocols, values should be calculated accordingly: layer functionality 1 physical (e.g., repeaters) 2 datalink/subnetwork (e.g., bridges) 3 internet (e.g., supports the IP) 4 end-to-end (e.g., supports the TCP) 7 applications (e.g., supports the SMP) For systems including OSI protocols, layers 5 and 6 may also be counted.
OID	{ system 7 }
NOTE	

OBJECT	sysORLastChange
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime at the time of the most recent change in state or value of any instance of sysORID
OID	{ system 8 }
NOTE	

OBJECT	sysORTable
SYNTAX	SEQUENCE OF SysOREntry
MAX-ACCESS	not-accessible

STATUS	current
DSCRPT	The (conceptual) table listing the capabilities of the local SNMPv2 entity acting in an agent role with respect to various MIB modules. SNMPv2 entities having dynamically-configurable support of MIB modules will have a dynamically-varying number of conceptual rows.
OID	{ system 9 }
NOTE	

OBJECT	sysOREntry
SYNTAX	SysOREntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An entry (conceptual row) in the sysORTable.
INDEX	sysORIndex
OID	{ sysORTable 1 }
NOTE	

OBJECT	sysORIndex
SYNTAX	INTEGER (1..2147483647)
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The auxiliary variable used for identifying instances of the columnar objects in the sysORTable.
OID	{ sysOREntry 1 }
NOTE	

OBJECT	sysORID
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-only
STATUS	current
DSCRPT	An authoritative identification of a capabilities statement with respect to various MIB modules supported by the local SNMPv2 entity acting in an agent role.
OID	{ sysOREntry 2 }
NOTE	

OBJECT	sysORDescr
SYNTAX	DisplayString
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A textual description of the capabilities identified by the corresponding instance of sysORID.

OID	{ sysOREntry 3 }
NOTE	

OBJECT	sysORUpTime
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime at the time this conceptual row was last instanciated.
OID	{ sysOREntry 4 }
NOTE	

OBJECT	snmpInPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of messages delivered to the SNMP entity from the transport service.
OID	{ snmp 1 }
NOTE	

OBJECT	snmpInBadVersions
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of SNMP messages which were delivered to the SNMP entity and were for an unsupported SNMP version.
OID	{ snmp 3 }
NOTE	

OBJECT	snmpInBadCommunityNames
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of SNMP messages delivered to the SNMP entity which used a SNMP community name not known to said entity.
OID	{ snmp 4 }
NOTE	

OBJECT	snmpInBadCommunityUses
SYNTAX	Counter32

MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of SNMP messages delivered to the SNMP entity which represented an SNMP operation which was not allowed by the SNMP community named in the message.
OID	{ snmp 5 }
NOTE	

OBJECT	snmpInASNParseErrs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of ASN.1 or BER errors encountered by the SNMP entity when decoding received SNMP messages.
OID	{ snmp 6 }
NOTE	

OBJECT	snmpEnableAuthenTraps
SYNTAX	INTEGER { enabled(1), disabled(2) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	Indicates whether the SNMP entity is permitted to generate authenticationFailure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authenticationFailure traps may be disabled. Note that it is strongly recommended that this object be stored in non-volatile memory so that it remains constant across re-initializations of the network management system.
OID	{ snmp 30 }
NOTE	

OBJECT	snmpSilentDrops
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity which were silently dropped because the size of a reply containing an alternate Response-PDU with an empty variable-bindings field was greater than either a local constraint or the maximum message size associated with the originator of the request.

OID	{ snmp 31 }
NOTE	

OBJECT	snmpProxyDrops
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity which were silently dropped because the transmission of the (possibly translated) message to a proxy target failed in a manner (other than a time-out) such that no Response-PDU could be returned.
OID	{ snmp 32 }
NOTE	

OBJECT	snmpTrapOID
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	accessible-for-notify
STATUS	current
DSCRPT	The authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2-Trap-PDU and InformRequest-PDU
OID	{ snmpTrap 1 }
NOTE	

OBJECT	snmpTrapEnterprise
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	accessible-for-notify
STATUS	current
DSCRPT	The authoritative identification of the enterprise associated with the trap currently being sent. When a SNMPv2 proxy agent is mapping an RFC1157 Trap-PDU into a SNMPv2-Trap-PDU, this variable occurs as the last varbind.
OID	{ snmpTrap 3 }
NOTE	

TRAP	coldStart
STATUS	current
DSCRPT	A coldStart trap signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself and that its configuration may have been altered.
OID	{ snmpTraps 1 }

NOTE	

TRAP	warmStart
STATUS	current
DSCRPT	A warmStart trap signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself such that its configuration is unaltered.
OID	{ snmpTraps 2 }
NOTE	

TRAP	authenticationFailure
STATUS	current
DSCRPT	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.
OID	{ snmpTraps 5 }
NOTE	

OBJECT	snmpSetSerialNo
SYNTAX	TestAndIncr
MAX-ACCESS	read-write
STATUS	current
DSCRPT	An advisory lock used to allow several cooperating SNMPv2 entities, all acting in a manager role, to coordinate their use of the SNMPv2 set operation. This object is used for coarse-grain coordination. To achieve fine-grain coordination, one or more similar objects might be defined within each MIB group, as appropriate.
OID	{ snmpSet 1 }
NOTE	

COMPLIANCE	snmpBasicCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement the SNMPv2 MIB.
MANDATORY-G	snmpGroup, snmpSetGroup, systemGroup, snmpBasicNotificationsGroup
GROUP	snmpCommunityGroup
DSCRPT	This group is mandatory for SNMPv2 entities which support community-based authentication.
OID	{ snmpMIBCompliances 2 }

NOTE	

OBJECT-G	snmpGroup
OBJECTS	snmpInPkts, snmpInBadVersions, snmpInASNParseErrs, snmpSilentDrops, snmpProxyDrops, snmpEnableAuthenTraps
STATUS	current
DSCRPT	A collection of objects providing basic instrumentation and control of an SNMPv2 entity.
CID	{ snmpMIBGroups 8 }
NOTE	

OBJECT-G	snmpCommunityGroup
OBJECTS	snmpInBadCommunityNames, snmpInBadCommunityUses
STATUS	current
DSCRPT	A collection of objects providing basic instrumentation of a SNMPv2 entity which supports community-based authentication.
CID	{ snmpMIBGroups 9 }
NOTE	

OBJECT-G	snmpSetGroup
OBJECTS	snmpSetSerialNo
STATUS	current
DSCRPT	A collection of objects which allow several cooperating SNMPv2 entities, all acting in a manager role, to coordinate their use of the SNMPv2 set operation.
CID	{ snmpMIBGroups 5 }
NOTE	

OBJECT-G	systemGroup
OBJECTS	sysDescr, sysObjectID, sysUpTime, sysContact, sysName, sysLocation, sysServices, sysORLastChange, sysORID, sysORUpTime, sysORDescr
STATUS	current
DSCRPT	The system group defines objects which are common to all managed systems.
CID	{ snmpMIBGroups 6 }
NOTE	

NOTIFICATION-G	snmpBasicNotificationsGroup
NOTIFICATIONS	coldStart, authenticationFailure
STATUS	current
DSCRPT	The two notifications which an SNMPv2 entity is required to implement.

OID	{ snmpMIBGroups 7 }
NOTE	

OBJECT	snmpOutPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Messages which were passed from the SNMP protocol entity to the transport service.
OID	{ snmp 2 }
NOTE	

OBJECT	snmpInTooBigs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is `tooBig'.
OID	{ snmp 8 }
NOTE	

OBJECT	snmpInNoSuchNames
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is `noSuchName'.
OID	{ snmp 9 }
NOTE	

OBJECT	snmpInBadValues
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is `badValue'.
OID	{ snmp 10 }

NOTE	

OBJECT	snmpInReadOnlys
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number valid SNMP PDUs which were 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 which 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.
OID	{ snmp 11 }
NOTE	

OBJECT	snmpInGenErrs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is `genErr'.
OID	{ snmp 12 }
NOTE	

OBJECT	snmpInTotalReqVars
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.
OID	{ snmp 13 }
NOTE	

OBJECT	snmpInTotalSetVars
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete

DSCRPT	The total number of MIB objects which have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.
OID	{ snmp 14 }
NOTE	

OBJECT	snmpInGetRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Request PDUs which have been accepted and processed by the SNMP protocol entity.
OID	{ snmp 15 }
NOTE	

OBJECT	snmpInGetNexts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Next PDUs which have been accepted and processed by the SNMP protocol entity.
OID	{ snmp 16 }
NOTE	

OBJECT	snmpInSetRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Set-Request PDUs which have been accepted and processed by the SNMP protocol entity.
OID	{ snmp 17 }
NOTE	

OBJECT	snmpInGetResponses
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Response PDUs which have been accepted and processed by the SNMP

	protocol entity.
OID	{ snmp 18 }
NOTE	

OBJECT	snmpInTraps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Trap PDUs which have been accepted and processed by the SNMP protocol entity.
OID	{ snmp 19 }
NOTE	

OBJECT	snmpOutTooBigs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is `tooBig.'
OID	{ snmp 20 }
NOTE	

OBJECT	snmpOutNoSuchNames
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status is `noSuchName'.
OID	{ snmp 21 }
NOTE	

OBJECT	snmpOutBadValues
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is

	`badValue'.
OID	{ snmp 22 }
NOTE	

OBJECT	snmpOutGenErrs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is `genErr'.
OID	{ snmp 24 }
NOTE	

OBJECT	snmpOutGetRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Request PDUs which have been generated by the SNMP protocol entity.
OID	{ snmp 25 }
NOTE	

OBJECT	snmpOutGetNexts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Next PDUs which have been generated by the SNMP protocol entity.
OID	{ snmp 26 }
NOTE	

OBJECT	snmpOutSetRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Set-Request PDUs which have been generated by the SNMP protocol entity.
OID	{ snmp 27 }
NOTE	

OBJECT	snmpOutGetResponses

SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Get-Response PDUs which have been generated by the SNMP protocol entity.
OID	{ snmp 28 }
NOTE	

OBJECT	snmpOutTraps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The total number of SNMP Trap PDUs which have been generated by the SNMP protocol entity.
OID	{ snmp 29 }
NOTE	

OBJECT-G	snmpObsoleteGroup
OBJECTS	snmpOutPkts, snmpInTooBigs, snmpInNoSuchNames, snmpInBadValues, snmpInReadOnlys, snmpInGenErrs, snmpInTotalReqVars, snmpInTotalSetVars, snmpInGetRequests, snmpInGetNexts, snmpInSetRequests, snmpInGetResponses, snmpInTraps, snmpOutTooBigs, snmpOutNoSuchNames, snmpOutBadValues, snmpOutGenErrs, snmpOutGetRequests, snmpOutGetNexts, snmpOutSetRequests, snmpOutGetResponses, snmpOutTraps
STATUS	obsolete
DSCRPT	A collection of objects from RFC 1213 made obsolete by this MIB.
OID	{ snmpMIBGroups 10 }
NOTE	

3.4 SNMPv2-TC

TC	DisplayString
STATUS	current
DSCRPT	Represents textual information taken from the NVT ASCII character set, as defined in pages 4, 10-11 of RFC 854. To summarize RFC 854, the NVT ASCII repertoire specifies: - the use of character codes 0-127 (decimal) - the graphics characters (32-126) are interpreted as US ASCII

	- NUL, LF, CR, BEL, BS, HT, VT and FF have the special meanings specified in RFC 854 - the other 25 codes have no standard interpretation - the sequence 'CR LF' means newline - the sequence 'CR NUL' means carriage-return - an 'LF' not preceded by a 'CR' means moving to the same column on the next line. - the sequence 'CR x' for any x other than LF or NUL is illegal. (Note that this also means that a string may end with either 'CR LF' or 'CR NUL', but not with CR.) Any object defined using this syntax may not exceed 255 characters in length.
SYNTAX	OCTET STRING (SIZE (0..255))

TC	PhysAddress
STATUS	current
DSCRPT	Represents media- or physical-level addresses.
SYNTAX	OCTET STRING

TC	MacAddress
STATUS	current
DSCRPT	Represents an 802 MAC address represented in the `canonical' order defined by IEEE 802.1a, i.e., as if it were transmitted least significant bit first, even though 802.5 (in contrast to other 802.x protocols) requires MAC addresses to be transmitted most significant bit first.
SYNTAX	OCTET STRING (SIZE (6))

TC	TruthValue
STATUS	current
DSCRPT	Represents a boolean value.
SYNTAX	INTEGER { true(1), false(2) }

TC	TestAndIncr
STATUS	current
DSCRPT	Represents integer-valued information used for atomic operations. When the management protocol is used to specify that an object instance having this syntax is to be modified, the new value supplied via the management protocol must precisely match the value presently held by the instance. If not, the management protocol set operation

	fails with an error of `inconsistentValue'. Otherwise, if the current value is the maximum value of 2^31-1 (2147483647 decimal), then the value held by the instance is wrapped to zero; otherwise, the value held by the instance is incremented by one. (Note that regardless of whether the management protocol set operation succeeds, the variable-binding in the request and response PDUs are identical.) The value of the ACCESS clause for objects having this syntax is either `read-write' or `read-create'. When an instance of a columnar object having this syntax is created, any value may be supplied via the management protocol. When the network management portion of the system is re-initialized, the value of every object instance having this syntax must either be incremented from its value prior to the re-initialization, or (if the value prior to the re-initialization is unknown) be set to a pseudo-randomly generated value.
SYNTAX	INTEGER (0..2147483647)

TC	AutonomousType
STATUS	current
DSCRPT	Represents an independently extensible type identification value. It may, for example, indicate a particular sub-tree with further MIB definitions, or define a particular type of protocol or hardware.
SYNTAX	OBJECT IDENTIFIER

TC	InstancePointer
STATUS	obsolete
DSCRPT	A pointer to either a specific instance of a MIB object or a conceptual row of a MIB table in the managed device. In the latter case, by convention, it is the name of the particular instance of the first accessible columnar object in the conceptual row The two uses of this textual convention are replaced by VariablePointer and RowPointer, respectively.
SYNTAX	OBJECT IDENTIFIER

TC	VariablePointer
STATUS	current
DSCRPT	A pointer to a specific object instance. For example, sysContact.0 or ifInOctets.3
SYNTAX	OBJECT IDENTIFIER

TC	RowPointer
STATUS	current
DSCRPT	Represents a pointer to a conceptual row The value is the name of the instance of the first accessible columnar object in the conceptual row For example, ifIndex.3 would point to the 3rd row in the ifTable (note that if ifIndex were not-accessible, then ifDescr.3 would be used instead).
SYNTAX	OBJECT IDENTIFIER

TC	RowStatus
STATUS	current
DSCRPT	The RowStatus textual convention is used to manage the creation and deletion of conceptual rows, and is used as the value of the SYNTAX clause for the status column of a conceptual row (as described in Section 7.7.1 of [2].) (snip) Conceptual Row Deletion For deletion of conceptual rows, a management protocol set operation is issued which sets the instance of the status column to `destroy'.  This request may be made regardless of the current value of the status column (e.g., it is possible to delete conceptual rows which are either `notReady', `notInService' or `active'.) If the operation succeeds, then all instances associated with the conceptual row are immediately removed.
SYNTAX	INTEGER { active(1), notInService(2), notReady(3), createAndGo(4), createAndWait(5), destroy(6) }

TC	TimeStamp
STATUS	current
DSCRPT	The value of the sysUpTime object at which a specific occurrence happened. The specific occurrence must be defined in the description of any object defined using this type. If sysUpTime is reset to zero as a result of a re-initialization of the network management (sub)system, then the values of all TimeStamp objects are also reset. However, after approximately 497 days without a re-initialization, the sysUpTime object will reach $2^{32}-1$ and then increment around to zero; in this case, existing values of TimeStamp objects do not change. This can lead to ambiguities in the value of TimeStamp objects.

SYNTAX	TimeTicks

TC	TimeInterval
STATUS	current
DSCRPT	A period of time, measured in units of 0.01 seconds.
SYNTAX	INTEGER (0..2147483647)

TC	DateAndTime
STATUS	current
DSCRPT	A date-time specification. field octets contents range 1 1-2 year* 0..65536 2 3 month 1..12 3 4 day 1..31 4 5 hour 0..23 5 6 minutes 0..59 6 7 seconds 0..60 (use 60 for leap-second) 7 8 deci-seconds 0..9 8 9 direction from UTC '+' / '-' 9 10 hours from UTC* 0..13 10 11 minutes from UTC 0..59 * Notes: - the value of year is in network-byte order - daylight saving time in New Zealand is +13 For example, Tuesday May 26, 1992 at 1:30.15 PM EDT would be displayed as: 1992-5-26, 13:30:15.0, -4:0 Note that if only local time is known, then timezone information (fields 8-10) is not present.
SYNTAX	OCTET STRING (SIZE (8 | 11))

TC	StorageType
STATUS	current
DSCRPT	Describes the memory realization of a conceptual row A row which is volatile(2) is lost upon reboot. A row which is either nonVolatile(3), permanent(4) or readOnly(5), is backed up by stable storage. A row which is permanent(4) can be changed but not deleted. A row which is readOnly(5) cannot be changed nor deleted. If the value of an object with this syntax is either permanent(4) or readOnly(5), it cannot be written.

	Conversely, if the value is either other(1), volatile(2) or nonVolatile(3), it cannot be modified to be permanent(4) or readOnly(5). (All illegal modifications result in a 'wrongValue' error.) Every usage of this textual convention is required to specify the columnar objects which a permanent(4) row must at a minimum allow to be writable.
SYNTAX	INTEGER { other(1), volatile(2), nonVolatile(3), permanent(4), readOnly(5) }

TC	TDomain
STATUS	current
DSCRPT	Denotes a kind of transport service. Some possible values, such as snmpUDPDomain, are defined in the SNMPv2-TM MIB module. Other possible values are defined in other MIB modules.
SYNTAX	OBJECT IDENTIFIER

TC	TAddress
STATUS	current
DSCRPT	Denotes a transport service address. A TAddress value is always interpreted within the context of a TDomain value. Thus, each definition of a TDomain value must be accompanied by a definition of a textual convention for use with that TDomain. Some possible textual conventions, such as SnmpUDPAddress for snmpUDPDomain, are defined in the SNMPv2-TM MIB module. Other possible textual conventions are defined in other MIB modules.
SYNTAX	OCTET STRING (SIZE (1..255))

3.5 SNMP-FRAMEWORK-MIB

MODULE	snmpFrameworkMIB
UPDATED	9901190000Z
DSCRPT	The SNMP Management Architecture MIB
REVISION	9901190000Z
DSCRPT	Updated editors' addresses, fixed typos. Published as RFC2571.
REVISION	9711200000Z
DSCRPT	The initial version, published in RFC 2271.
OID	{ snmpModules 10 }
NOTE	

TC	SnmpEngineID
STATUS	current
DSCRPT	An SNMP engine's administratively-unique identifier. Objects of this type are for identification, not for addressing, even though it is possible that an address may have been used in the generation of a specific value.
	The value for this object may not be all zeros or all 'ff'H or the empty (zero length) string.
	The initial value for this object may be configured via an operator console entry or via an algorithmic function. In the latter case, the following example algorithm is recommended.
	In cases where there are multiple engines on the same system, the use of this algorithm is NOT appropriate, as it would result in all of those engines ending up with the same ID value.
	1) The very first bit is used to indicate how the rest of the data is composed.
	0 - as defined by enterprise using former methods that existed before SNMPv3. See item 2 below
	1 - as defined by this architecture, see item 3 below
	Note that this allows existing uses of the engineID (also known as AgentID [RFC1910]) to co-exist with any new uses.
	2) The snmpEngineID has a length of 12 octets.
	The first four octets are set to the binary equivalent of the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). For example, if Acme Networks has been assigned { enterprises 696 }, the first four octets would be assigned '000002b8'H
	The remaining eight octets are determined via one or more enterprise-specific methods. Such methods must be designed so as to maximize the possibility that the value of this object will be unique in the agent's administrative domain.

	For example, it may be the IP address of the SNMP entity, or the MAC address of one of the interfaces, with each address suitably padded with random octets. If multiple methods are defined, then it is recommended that the first octet indicate the method being used and the remaining octets be a function of the method. 3) The length of the octet strings varies. The first four octets are set to the binary equivalent of the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). For example, if Acme Networks has been assigned { enterprises 696 }, the first four octets would be assigned '000002b8'H The very first bit is set to 1. For example, the above value for Acme Networks now changes to be '800002b8'H The fifth octet indicates how the rest (6th and following octets) are formatted. The values for the fifth octet are: 0 - reserved, unused. 1 - IPv4 address (4 octets) lowest non-special IP address 2 - IPv6 address (16 octets) lowest non-special IP address 3 - MAC address (6 octets) lowest IEEE MAC address, canonical order 4 - Text, administratively assigned Maximum remaining length 27 5 - Octets, administratively assigned Maximum remaining length 27 6-127 - reserved, unused 127-255 - as defined by the enterprise Maximum remaining length 27
SYNTAX	OCTET STRING (SIZE(5..32))

TC	SnmpSecurityModel
STATUS	current
DSCRPT	An identifier that uniquely identifies asecurityModel of the Security Subsystem within the SNMP Management Architecture. The values for securityModel are allocated as follows: - The zero value is reserved. - Values between 1 and 255, inclusive, are reserved for standards-track Security Models and are managed by the Internet Assigned Numbers Authority (IANA). - Values greater than 255 are allocated to enterprise-specific Security Models. An enterprise-specific securityModel value is defined to be: enterpriseID * 256 + security model within enterprise For example, the fourth Security Model defined by the enterprise whose enterpriseID is 1 would be 260. This scheme for allocation of securityModel values allows for a maximum of 255 standards-based Security Models, and for a maximum of 255 Security Models per enterprise. It is believed that the assignment of new securityModel values will be rare in practice because the larger the number of simultaneously utilized Security Models, the larger the chance that interoperability will suffer. Consequently, it is believed that such a range will be sufficient. In the unlikely event that the standards committee finds this number to be insufficient over time, an enterprise number can be allocated to obtain an additional 255 possible values. Note that the most significant bit must be zero; hence, there are 23 bits allocated for various organizations to design and define non-standard securityModels. This limits the ability to define new proprietary implementations of Security Models to the first 8,388,608 enterprises.

	It is worthwhile to note that, in its encoded form, the securityModel value will normally require only a single byte since, in practice, the leftmost bits will be zero for most messages and sign extension is suppressed by the encoding rules. As of this writing, there are several values of securityModel defined for use with SNMP or reserved for use with supporting MIB objects. They are as follows: 0 reserved for 'any' 1 reserved for SNMPv1 2 reserved for SNMPv2c 3 User-Based Security Model (USM)
SYNTAX	INTEGER(0 .. 2147483647)

TC	SnmpMessageProcessingModel
STATUS	current
DSCRPT	An identifier that uniquely identifies a MessageProcessing Model of the Message Processing Subsystem within a SNMP Management Architecture. The values for messageProcessingModel are allocated as follows: - Values between 0 and 255, inclusive, are reserved for standards-track Message Processing Models and are managed by the Internet Assigned Numbers Authority (IANA). - Values greater than 255 are allocated to enterprise-specific Message Processing Models. An enterprise messageProcessingModel value is defined to be: enterpriseID * 256 + messageProcessingModel within enterprise For example, the fourth Message Processing Model defined by the enterprise whose enterpriseID is 1 would be 260. This scheme for allocating messageProcessingModel values allows for a maximum of 255 standards-based Message Processing Models, and for a maximum of 255 Message Processing Models per

	enterprise. It is believed that the assignment of new messageProcessingModel values will be rare in practice because the larger the number of simultaneously utilized Message Processing Models, the larger the chance that interoperability will suffer. It is believed that such a range will be sufficient.　In the unlikely event that the standards committee finds this number to be insufficient over time, an enterprise number can be allocated to obtain an additional 256 possible values. Note that the most significant bit must be zero; hence, there are 23 bits allocated for various organizations to design and define non-standard messageProcessingModels.　This limits the ability to define new proprietary implementations of Message Processing Models to the first 8,388,608 enterprises. It is worthwhile to note that, in its encoded form, the messageProcessingModel value will normally require only a single byte since, in practice, the leftmost bits will be zero for most messages and sign extension is suppressed by the encoding rules. As of this writing, there are several values of messageProcessingModel defined for use with SNMP. They are as follows: 0　reserved for SNMPv1 1　reserved for SNMPv2c 2　reserved for SNMPv2u and SNMPv2* 3　reserved for SNMPv3
SYNTAX	INTEGER(0 .. 2147483647)

TC	SnmpSecurityLevel
STATUS	current
DSCRPT	A Level of Security at which SNMP messages can besent or with which operations are being processed; in particular, one of: noAuthNoPriv - without authentication and without privacy, authNoPriv　 - with authentication but without privacy,

	authPriv - with authentication and with privacy. These three values are ordered such that noAuthNoPriv is less than authNoPriv and authNoPriv is less than authPriv.
SYNTAX	INTEGER { noAuthNoPriv(1), authNoPriv(2), authPriv(3) }

TC	SnmpAdminString
STATUS	current
DSCRPT	An octet string containing administrative information, preferably in human-readable form. To facilitate internationalization, this information is represented using the ISO/IEC IS 10646-1 character set, encoded as an octet string using the UTF-8 transformation format described in [RFC2279]. Since additional code points are added by amendments to the 10646 standard from time to time, implementations must be prepared to encounter any code point from 0x00000000 to 0x7fffffff. Byte sequences that do not correspond to the valid UTF-8 encoding of a code point or are outside this range are prohibited. The use of control codes should be avoided. When it is necessary to represent a newline, the control code sequence CR LF should be used. The use of leading or trailing white space should be avoided. For code points not directly supported by user interface hardware or software, an alternative means of entry and display, such as hexadecimal, may be provided. For information encoded in 7-bit US-ASCII, the UTF-8 encoding is identical to the US-ASCII encoding. UTF-8 may require multiple bytes to represent a single character / code point; thus the length of this object in octets may be different from the number of characters encoded. Similarly,

size constraints refer to the number of encoded
octets, not the number of characters represented
by an encoding.

Note that when this TC is used for an object that
is used or envisioned to be used as an index, then
a SIZE restriction MUST be specified so that the
number of sub-identifiers for any object instance
does not exceed the limit of 128, as defined by
[RFC1905].

Note that the size of an SnmpAdminString object is
measured in octets, not characters.

SYNTAX	OCTET STRING (SIZE (0..255))

OBJECT	snmpEngineID
SYNTAX	SnmpEngineID
MAX-ACCESS	read-only
STATUS	current
DSCRPT	An SNMP engine's administratively-unique identifier.
CID	{ snmpEngine 1 }
NOTE	

OBJECT	snmpEngineBoots
SYNTAX	INTEGER (1..2147483647)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of times that the SNMP engine has (re-)initialized itself since snmpEngineID was last configured.
CID	{ snmpEngine 2 }
NOTE	

OBJECT	snmpEngineTime
SYNTAX	INTEGER (0..2147483647)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of seconds since the value of the snmpEngineBoots object last changed. When incrementing this object's value would cause it to exceed its maximum, snmpEngineBoots is incremented as if a re-initialization had occurred, and this object's value consequently reverts to zero.
CID	{ snmpEngine 3 }

NOTE	

OBJECT	snmpEngineMaxMessageSize
SYNTAX	INTEGER (484..2147483647)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The maximum length in octets of an SNMP message which this SNMP engine can send or receive and process, determined as the minimum of the maximum message size values supported among all of the transports available to and supported by the engine.
OID	{ snmpEngine 4 }
NOTE	

OBJECT-ID	snmpAuthProtocols
STATUS	current
DSCRPT	Registration point for standards-track authentication protocols used in SNMP Management Frameworks.
OID	{ snmpFrameworkAdmin 1 }
NOTE	

OBJECT-ID	snmpPrivProtocols
STATUS	current
DSCRPT	Registration point for standards-track privacy protocols used in SNMP Management Frameworks.
OID	{ snmpFrameworkAdmin 2 }
NOTE	

COMPLIANCE	snmpFrameworkMIBCompliance
STATUS	current
DSCRPT	The compliance statement for SNMP engines which implement the SNMP Management Framework MIB.
MANDATORY-G	snmpEngineGroup
OID	{ snmpFrameworkMIBCompliances 1 }
NOTE	

OBJECT-G	snmpEngineGroup
OBJECTS	snmpEngineID, snmpEngineBoots, snmpEngineTime, snmpEngineMaxMessageSize
STATUS	current
DSCRPT	A collection of objects for identifying and determining the configuration and current timeliness values of an SNMP engine.

CID	{ snmpFrameworkMIBGroups 1 }
NOTE	

3.6 IP-MIB

MODULE	ipMIB
UPDATED	9411010000Z
DSCRPT	The MIB module for managing IP and ICMP implementations, but excluding their management of IP routes.
REVISION	9103310000Z
DSCRPT	The initial revision of this MIB module was part of MIB-II.
CID	{ mib-2 48}
NOTE	

OBJECT	ipForwarding
SYNTAX	INTEGER { forwarding(1), notForwarding(2) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The indication of whether this entity is acting as an IP router in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP routers forward datagrams. IP hosts do not (except those source-routed via the host).
CID	{ ip 1 }
NOTE	

OBJECT	ipDefaultTTL
SYNTAX	INTEGER (1..255)
MAX-ACCESS	read-write
STATUS	current
DSCRPT	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.
CID	{ ip 2 }
NOTE	

OBJECT	ipInReceives
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The total number of input datagrams received from interfaces, including those received in error.
OID	{ ip 3 }
NOTE	

OBJECT	ipInHdrErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.
OID	{ ip 4 }
NOTE	

OBJECT	ipInAddrErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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 (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
OID	{ ip 5 }
NOTE	

OBJECT	ipForwDatagrams
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP routers, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route option processing was successful.
OID	{ ip 6 }
NOTE	

OBJECT	ipInUnknownProtos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
OID	{ ip 7 }
NOTE	

OBJECT	ipInDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.
OID	{ ip 8 }
NOTE	

OBJECT	ipInDelivers
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).
OID	{ ip 9 }
NOTE	

OBJECT	ipOutRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams.
OID	{ ip 10 }
NOTE	

OBJECT	ipOutDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.
OID	{ ip 11 }
NOTE	

OBJECT	ipOutNoRoutes
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this `no-route' criterion. Note that this includes any datagrams which a host cannot route because all of its default routers are down.
OID	{ ip 12 }
NOTE	

OBJECT	ipReasmTimeout
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.
OID	{ ip 13 }
NOTE	

OBJECT	ipReasmReqds
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IP fragments received which needed to be reassembled at this entity.
OID	{ ip 14 }
NOTE	

OBJECT	ipReasmOKs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IP datagrams successfully re-assembled.
OID	{ ip 15 }
NOTE	

OBJECT	ipReasmFails
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of failures detected by the IP re-assembly algorithm(for whatever reason: timed out, errors, etc). Note that 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.
OID	{ ip 16 }
NOTE	

OBJECT	ipFragOKs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IP datagrams that have been successfully fragmented at this entity.
OID	{ ip 17 }
NOTE	

OBJECT	ipFragFails
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set.
OID	{ ip 18 }
NOTE	

OBJECT	ipFragCreates
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.
CID	{ ip 19 }
NOTE	

OBJECT	ipAddrTable
SYNTAX	SEQUENCE OF IpAddrEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The table of addressing information relevant to this entity's IP addresses.
CID	{ ip 20 }
NOTE	

OBJECT	ipAddrEntry
SYNTAX	IpAddrEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The addressing information for one of this entity's IP addresses.
INDEX	ipAdEntAddr
CID	{ ipAddrTable 1 }
NOTE	

OBJECT	ipAdEntAddr
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The IP address to which this entry's addressing information pertains.
CID	{ ipAddrEntry 1 }
NOTE	

OBJECT	ipAdEntIfIndex
SYNTAX	INTEGER (1..2147483647)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of RFC 1573's ifIndex.
CID	{ ipAddrEntry 2 }

NOTE	

OBJECT	ipAdEntNetMask
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ ipAddrEntry 3 }
NOTE	

OBJECT	ipAdEntBcastAddr
SYNTAX	INTEGER (0..1)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ ipAddrEntry 4 }
NOTE	

OBJECT	ipAdEntReasmMaxSize
SYNTAX	INTEGER (0..65535)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.
OID	{ ipAddrEntry 5 }
NOTE	

OBJECT	ipNetToMediaTable
SYNTAX	SEQUENCE OF IpNetToMediaEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The IP Address Translation table used for mapping from IP addresses to physical addresses.
OID	{ ip 22 }

NOTE	

OBJECT	ipNetToMediaEntry
SYNTAX	IpNetToMediaEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	Each entry contains one IpAddress to `physical' address equivalence.
INDEX	ipNetToMediaIfIndex, ipNetToMediaNetAddress
OID	{ ipNetToMediaTable 1 }
NOTE	

OBJECT	ipNetToMediaIfIndex
SYNTAX	INTEGER (1..2147483647)
MAX-ACCESS	read-create
STATUS	current
DSCRPT	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 RFC 1573's ifIndex.
OID	{ ipNetToMediaEntry 1 }
NOTE	

OBJECT	ipNetToMediaPhysAddress
SYNTAX	PhysAddress
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The media-dependent `physical' address.
OID	{ ipNetToMediaEntry 2 }
NOTE	

OBJECT	ipNetToMediaNetAddress
SYNTAX	IpAddress
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The IpAddress corresponding to the media-dependent `physical' address.
OID	{ ipNetToMediaEntry 3 }
NOTE	

OBJECT	ipNetToMediaType
SYNTAX	INTEGER { other(1), invalid(2), dynamic(3), static(4) }

MAX-ACCESS	read-create
STATUS	current
DSCRPT	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 said entry from the mapping identified with said 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.
OID	{ ipNetToMediaEntry 4 }
NOTE	

OBJECT	ipRoutingDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of routing entries which 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.
OID	{ ip 23 }
NOTE	

OBJECT	icmpInMsgs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of ICMP messages which the entity received.　Note that this counter includes all those counted by icmpInErrors.
OID	{ icmp 1 }
NOTE	

OBJECT	icmpInErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).

OID	{ icmp 2 }
NOTE	

OBJECT	icmpInDestUnreachs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Destination Unreachable messages received.
OID	{ icmp 3 }
NOTE	

OBJECT	icmpInTimeExcds
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Time Exceeded messages received.
OID	{ icmp 4 }
NOTE	

OBJECT	icmpInParmProbs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Parameter Problem messages received.
OID	{ icmp 5 }
NOTE	

OBJECT	icmpInSrcQuenchs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Source Quench messages received.
OID	{ icmp 6 }
NOTE	

OBJECT	icmpInRedirects
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Redirect messages received.
OID	{ icmp 7 }

NOTE	

OBJECT	icmpInEchos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo (request) messages received.
OID	{ icmp 8 }
NOTE	

OBJECT	icmpInEchoReps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo Reply messages received.
OID	{ icmp 9 }
NOTE	

OBJECT	icmpInTimestamps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Timestamp (request) messages received.
OID	{ icmp 10 }
NOTE	

OBJECT	icmpInTimestampReps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Timestamp Reply messages received.
OID	{ icmp 11 }
NOTE	

OBJECT	icmpInAddrMasks
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Address Mask Request messages received.
OID	{ icmp 12 }
NOTE	

OBJECT	icmpInAddrMaskReps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Address Mask Reply messages received.
OID	{ icmp 13 }
NOTE	

OBJECT	icmpOutMsgs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.
OID	{ icmp 14 }
NOTE	

OBJECT	icmpOutErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP messages which 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 error which contribute to this counter's value.
OID	{ icmp 15 }
NOTE	

OBJECT	icmpOutDestUnreachs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Destination Unreachable messages sent.
OID	{ icmp 16 }
NOTE	

OBJECT	icmpOutTimeExcds
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The number of ICMP Time Exceeded messages sent.
OID	{ icmp 17 }
NOTE	

OBJECT	icmpOutParmProbs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Parameter Problem messages sent.
OID	{ icmp 18 }
NOTE	

OBJECT	icmpOutSrcQuenchs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Source Quench messages sent.
OID	{ icmp 19 }
NOTE	

OBJECT	icmpOutRedirects
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.
OID	{ icmp 20 }
NOTE	

OBJECT	icmpOutEchos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo (request) messages sent.
OID	{ icmp 21 }
NOTE	

OBJECT	icmpOutEchoReps
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The number of ICMP Echo Reply messages sent.
OID	{ icmp 22 }
NOTE	

OBJECT	icmpOutTimestamps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Timestamp (request) messages sent.
OID	{ icmp 23 }
NOTE	

OBJECT	icmpOutTimestampReps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Timestamp Reply messages sent.
OID	{ icmp 24 }
NOTE	

OBJECT	icmpOutAddrMasks
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Address Mask Request messages sent.
OID	{ icmp 25 }
NOTE	

OBJECT	icmpOutAddrMaskReps
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Address Mask Reply messages sent.
OID	{ icmp 26 }
NOTE	

COMPLIANCE	ipMIBCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement IP.
MANDATORY-G	ipGroup, icmpGroup

OID	{ ipMIBCompliances 1 }
NOTE	

OBJECT-G	ipGroup
OBJECTS	ipForwarding, ipDefaultTTL, ipInReceives, ipInHdrErrors, ipInAddrErrors, ipForwDatagrams, ipInUnknownProtos, ipInDiscards, ipInDelivers, ipOutRequests, ipOutDiscards, ipOutNoRoutes, ipReasmTimeout, ipReasmReqds, ipReasmOKs, ipReasmFails, ipFragOKs, ipFragFails, ipFragCreates, ipAdEntAddr, ipAdEntIfIndex, ipAdEntNetMask, ipAdEntBcastAddr, ipAdEntReasmMaxSize, ipNetToMediaIfIndex, ipNetToMediaPhysAddress, ipNetToMediaNetAddress, ipNetToMediaType, ipRoutingDiscards
STATUS	current
DSCRPT	The ip group of objects providing for basic management of IP entities, exclusive of the management of IP routes.
OID	{ ipMIBGroups 1 }
NOTE	

OBJECT-G	icmpGroup
OBJECTS	icmpInMsgs, icmpInErrors, icmpInDestUnreachs, icmpInTimeExcds, icmpInParmProbs, icmpInSrcQuenchs, icmpInRedirects, icmpInEchos, icmpInEchoReps, icmpInTimestamps, icmpInTimestampReps, icmpInAddrMasks, icmpInAddrMaskReps, icmpOutMsgs, icmpOutErrors, icmpOutDestUnreachs, icmpOutTimeExcds, icmpOutParmProbs, icmpOutSrcQuenchs, icmpOutRedirects, icmpOutEchos, icmpOutEchoReps, icmpOutTimestamps, icmpOutTimestampReps, icmpOutAddrMasks, icmpOutAddrMaskReps
STATUS	current
DSCRPT	The icmp group of objects providing ICMP statistics.
OID	{ ipMIBGroups 2 }
NOTE	

3.7 TCP-MIB

MODULE	tcpMIB
UPDATED	9411010000Z
DSCRPT	The MIB module for managing TCP implementations.
REVISION	9103310000Z
DSCRPT	The initial revision of this MIB module was part of MIB-II.
OID	{ mib-2 49 }
NOTE	

OBJECT	tcpRtoAlgorithm

SYNTAX	INTEGER { other(1), constant(2), rsre(3), vanj(4) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The algorithm used to determine the timeout value used for retransmitting unacknowledged octets.
OID	{ tcp 1 }
NOTE	

OBJECT	tcpRtoMin
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ tcp 2 }
NOTE	

OBJECT	tcpRtoMax
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ tcp 3 }
NOTE	

OBJECT	tcpMaxConn
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.

CID	{ tcp 4 }
NOTE	

OBJECT	tcpActiveOpens
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.
CID	{ tcp 5 }
NOTE	

OBJECT	tcpPassiveOpens
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
CID	{ tcp 6 }
NOTE	

OBJECT	tcpAttemptFails
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
CID	{ tcp 7 }
NOTE	

OBJECT	tcpEstabResets
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
CID	{ tcp 8 }
NOTE	

OBJECT	tcpCurrEstab
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.
OID	{ tcp 9 }
NOTE	

OBJECT	tcpInSegs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of segments received, including those received in error. This count includes segments received on currently established connections.
OID	{ tcp 10 }
NOTE	

OBJECT	tcpOutSegs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
OID	{ tcp 11 }
NOTE	

OBJECT	tcpRetransSegs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of segments retransmitted - that is, the number of TCP segments transmitted containing one or more previously transmitted octets.
OID	{ tcp 12 }
NOTE	

OBJECT	tcpConnTable
SYNTAX	SEQUENCE OF TcpConnEntry
MAX-ACCESS	not-accessible
STATUS	current

DSCRPT	A table containing TCP connection-specific information.
OID	{ tcp 13 }
NOTE	

OBJECT	tcpConnEntry
SYNTAX	TcpConnEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A conceptual row of the tcpConnTable containing information about a particular current TCP connection. Each row of this table is transient, in that it ceases to exist when (or soon after) the connection makes the transition to the CLOSED state.
INDEX	tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort
OID	{ tcpConnTable 1 }
NOTE	

OBJECT	tcpConnState
SYNTAX	INTEGER { closed(1), listen(2), synSent(3), synReceived(4), established(5), finWait1(6), finWait2(7), closeWait(8), lastAck(9), closing(10), timeWait(11), deleteTCB(12) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The state of this TCP connection. The only value which 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, resulting in immediate termination of the connection. As an implementation-specific option, a RST segment may be sent from the managed node to the other TCP endpoint (note however that RST segments are not sent reliably).
OID	{ tcpConnEntry 1 }
NOTE	

OBJECT	tcpConnLocalAddress
SYNTAX	IpAddress

MAX-ACCESS	read-only
STATUS	current
DSCRPT	The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.
OID	{ tcpConnEntry 2 }
NOTE	

OBJECT	tcpConnLocalPort
SYNTAX	INTEGER (0..65535)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The local port number for this TCP connection.
OID	{ tcpConnEntry 3 }
NOTE	

OBJECT	tcpConnRemAddress
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The remote IP address for this TCP connection.
OID	{ tcpConnEntry 4 }
NOTE	

OBJECT	tcpConnRemPort
SYNTAX	INTEGER (0..65535)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The remote port number for this TCP connection.
OID	{ tcpConnEntry 5 }
NOTE	

OBJECT	tcpInErrs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of segments received in error (e.g., bad TCP checksums).
OID	{ tcp 14 }
NOTE	

OBJECT	tcpOutRsts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of TCP segments sent containing the RST flag.
OID	{ tcp 15 }
NOTE	

COMPLIANCE	tcpMIBCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement TCP.
MANDATORY-G	tcpGroup
OID	{ tcpMIBCompliances 1 }
NOTE	

OBJECT-G	tcpGroup
OBJECTS	tcpRtoAlgorithm, tcpRtoMin, tcpRtoMax, tcpMaxConn, tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpCurrEstab, tcpInSegs, tcpOutSegs, tcpRetransSegs, tcpConnState, tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort, tcpInErrs, tcpOutRsts
STATUS	current
DSCRPT	The tcp group of objects providing for management of TCP entities.
OID	{ tcpMIBGroups 1 }
NOTE	

3.8 UDP-MIB

MODULE	udpMIB
UPDATED	9411010000Z
DSCRPT	The MIB module for managing UDP implementations.
REVISION	9103310000Z
DSCRPT	The initial revision of this MIB module was part of MIB-II.
OID	{ mib-2 50 }
NOTE	

OBJECT	udpInDatagrams
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The total number of UDP datagrams delivered to UDP users.
OID	{ udp 1 }
NOTE	

OBJECT	udpNoPorts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of received UDP datagrams for which there was no application at the destination port.
OID	{ udp 2 }
NOTE	

OBJECT	udpInErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
OID	{ udp 3 }
NOTE	

OBJECT	udpOutDatagrams
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of UDP datagrams sent from this entity.
OID	{ udp 4 }
NOTE	

OBJECT	udpTable
SYNTAX	SEQUENCE OF UdpEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A table containing UDP listener information.
OID	{ udp 5 }
NOTE	

OBJECT	udpEntry
SYNTAX	UdpEntry

MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	Information about a particular current UDP listener.
INDEX	udpLocalAddress, udpLocalPort
OID	{ udpTable 1 }
NOTE	

OBJECT	udpLocalAddress
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The local IP address for this UDP listener. In the case of a UDP listener which is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.
OID	{ udpEntry 1 }
NOTE	

OBJECT	udpLocalPort
SYNTAX	INTEGER (0..65535)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The local port number for this UDP listener.
OID	{ udpEntry 2 }
NOTE	

COMPLIANCE	udpMIBCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement UDP.
MANDATORY-G	udpGroup
OID	{ udpMIBCompliances 1 }
NOTE	

OBJECT-G	udpGroup
OBJECTS	udpInDatagrams, udpNoPorts, udpInErrors, udpOutDatagrams, udpLocalAddress, udpLocalPort
STATUS	current
DSCRPT	The udp group of objects providing for management of UDP entities.
OID	{ udpMIBGroups 1 }
NOTE	

3.9 IP-FORWARD-MIB

MODULE	ipForward
UPDATED	9609190000Z
DSCRPT	The MIB module for the display of CIDR multipath IP Routes.
REVISION	9609190000Z
DSCRPT	Revisions made by the OSPF WG.
OID	{ ip 24 }
NOTE	

OBJECT	ipCidrRouteNumber
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of current ipCidrRouteTable entries that are not invalid.
OID	{ ipForward 3 }
NOTE	

OBJECT	ipCidrRouteTable
SYNTAX	SEQUENCE OF IpCidrRouteEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	This entity's IP Routing table.
OID	{ ipForward 4 }
NOTE	

OBJECT	ipCidrRouteEntry
SYNTAX	IpCidrRouteEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A particular route to a particular destination, under a particular policy.
INDEX	ipCidrRouteDest, ipCidrRouteMask, ipCidrRouteTos, ipCidrRouteNextHop
OID	{ ipCidrRouteTable 1 }
NOTE	

OBJECT	ipCidrRouteDest
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The destination IP address of this route.

This object may not take a Multicast (Class D)
address value.

Any assignment (implicit or otherwise) of an
instance of this object to a value x must be
rejected if the bitwise logical-AND of x with
the value of the corresponding instance of the
ipCidrRouteMask object is not equal to x.

CID	{ ipCidrRouteEntry 1 }
NOTE	

OBJECT	ipCidrRouteMask
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipCidrRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipCidrRouteMask by reference to the IP Address Class. Any assignment (implicit or otherwise) of an instance of this object to a value x must be rejected if the bitwise logical-AND of x with the value of the corresponding instance of the ipCidrRouteDest object is not equal to ipCidrRoute-Dest.
CID	{ ipCidrRouteEntry 2 }
NOTE	

OBJECT	ipCidrRouteTos
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The policy specifier is the IP TOS Field. The encoding of IP TOS is as specified by the following convention. Zero indicates the default path if no more specific policy applies.
	
	`+----+----+----+----+----+----+----+----+`
	`|         |                    |    |`
	`| PRECEDENCE |   TYPE OF SERVICE   | 0  |`
	`|         |                    |    |`
	`+----+----+----+----+----+----+----+----+`

	IP TOS Field Policy Contents Code 0 0 0 0 ==> 0 0 0 1 0 ==> 4 0 1 0 0 ==> 8 0 1 1 0 ==> 12 1 0 0 0 ==> 16 1 0 1 0 ==> 20 1 1 0 0 ==> 24 1 1 1 0 ==> 28 IP TOS Field Policy Contents Code 0 0 0 1 ==> 2 0 0 1 1 ==> 6 0 1 0 1 ==> 10 0 1 1 1 ==> 14 1 0 0 1 ==> 18 1 0 1 1 ==> 22 1 1 0 1 ==> 26 1 1 1 1 ==> 30
OID	{ ipCidrRouteEntry 3 }
NOTE	

OBJECT	ipCidrRouteNextHop
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	On remote routes, the address of the next sys- tem en route; Otherwise, 0.0.0.0.
OID	{ ipCidrRouteEntry 4 }
NOTE	

OBJECT	ipCidrRouteIfIndex
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The ifIndex value which identifies the local interface through which the next hop of this route should be reached.
OID	{ ipCidrRouteEntry 5 }
NOTE	

OBJECT	ipCidrRouteType
SYNTAX	INTEGER { other (1), reject (2), local (3), remote (4) }
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The type of route. Note that local(3) refers to a route for which the next hop is the final destination; remote(4) refers to a route for which the next hop is not the final destina- tion. Routes which do not result in traffic forwarding or

	rejection should not be displayed even if the implementation keeps them stored internally. reject (2) refers to a route which, if matched, discards the message as unreachable. This is used in some protocols as a means of correctly aggregating routes.
OID	{ ipCidrRouteEntry 6 }
NOTE	

OBJECT	ipCidrRouteProto
SYNTAX	INTEGER { other (1), local (2), netmgmt (3), icmp (4), egp (5), ggp (6), hello (7), rip (8), isIs (9), esIs (10), ciscoIgrp (11), bbnSpfIgp (12), ospf (13), bgp (14), idpr (15), ciscoEigrp (16) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.
OID	{ ipCidrRouteEntry 7 }
NOTE	

OBJECT	ipCidrRouteAge
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of `too old' can be implied except through knowledge of the routing protocol by which the route was learned.
OID	{ ipCidrRouteEntry 8 }
NOTE	

OBJECT	ipCidrRouteInfo
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-create
STATUS	current
DSCRPT	A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipCidrRouteProto value. If this information is not present, its value

	should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identif- ier, and any implementation conforming to ASN.1 and the Basic Encoding Rules must be able to generate and recognize this value.
OID	{ ipCidrRouteEntry 9 }
NOTE	

OBJECT	ipCidrRouteNextHopAS
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The Autonomous System Number of the Next Hop. The semantics of this object are determined by the routing-protocol specified in the route's ipCidrRouteProto value. When this object is unknown or not relevant its value should be set to zero.
OID	{ ipCidrRouteEntry 10 }
NOTE	

OBJECT	ipCidrRouteMetric1
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipCidrRouteProto value. If this metric is not used, its value should be set to -1.
OID	{ ipCidrRouteEntry 11 }
NOTE	

OBJECT	ipCidrRouteMetric2
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipCidrRouteProto value. If this metric is not used, its value should be set to -1.
OID	{ ipCidrRouteEntry 12 }
NOTE	

OBJECT	ipCidrRouteMetric3
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipCidrRouteProto value. If this metric is not used, its value should be set to -1.
OID	{ ipCidrRouteEntry 13 }
NOTE	

OBJECT	ipCidrRouteMetric4
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipCidrRouteProto value. If this metric is not used, its value should be set to -1.
OID	{ ipCidrRouteEntry 14 }
NOTE	

OBJECT	ipCidrRouteMetric5
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	current
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipCidrRouteProto value. If this metric is not used, its value should be set to -1.
OID	{ ipCidrRouteEntry 15 }
NOTE	

OBJECT	ipCidrRouteStatus
SYNTAX	RowStatus
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The row status variable, used according to row installation and removal conventions.
OID	{ ipCidrRouteEntry 16 }

NOTE	

COMPLIANCE	ipForwardCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement the ipForward MIB.
MANDATORY-G	ipForwardCidrRouteGroup
OID	{ ipForwardCompliances 1 }
NOTE	

OBJECT-G	ipForwardCidrRouteGroup
OBJECTS	ipCidrRouteNumber, ipCidrRouteDest, ipCidrRouteMask, ipCidrRouteTos, ipCidrRouteNextHop, ipCidrRouteIfIndex, ipCidrRouteType, ipCidrRouteProto, ipCidrRouteAge, ipCidrRouteInfo, ipCidrRouteNextHopAS, ipCidrRouteMetric1, ipCidrRouteMetric2, ipCidrRouteMetric3, ipCidrRouteMetric4, ipCidrRouteMetric5, ipCidrRouteStatus
STATUS	current
DSCRPT	The CIDR Route Table.
OID	{ ipForwardGroups 3 }
NOTE	

OBJECT	ipForwardNumber
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The number of current ipForwardTable entries that are not invalid.
OID	{ ipForward 1 }
NOTE	

OBJECT	ipForwardTable
SYNTAX	SEQUENCE OF IpForwardEntry
MAX-ACCESS	not-accessible
STATUS	obsolete
DSCRPT	This entity's IP Routing table.
OID	{ ipForward 2 }
NOTE	

OBJECT	ipForwardEntry
SYNTAX	IpForwardEntry
MAX-ACCESS	not-accessible
STATUS	obsolete

DSCRPT	A particular route to a particular destination, under a particular policy.
INDEX	ipForwardDest, ipForwardProto, ipForwardPolicy, ipForwardNextHop
OID	{ ipForwardTable 1 }
NOTE	

OBJECT	ipForwardDest
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route.

This object may not take a Multicast (Class D) address value.

Any assignment (implicit or otherwise) of an instance of this object to a value x must be rejected if the bitwise logical-AND of x with the value of the corresponding instance of the ipForwardMask object is not equal to x. |
| OID | { ipForwardEntry 1 } |
| NOTE | |

OBJECT	ipForwardMask
SYNTAX	IpAddress
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipForwardDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipForwardMask by reference to the IP Address Class.

Any assignment (implicit or otherwise) of an instance of this object to a value x must be rejected if the bitwise logical-AND of x with the value of the corresponding instance of the ipForwardDest object is not equal to ipForwardDest. |
| OID | { ipForwardEntry 2 } |
| NOTE | |

OBJECT	ipForwardPolicy
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	obsolete

DSCRPT	The general set of conditions that would cause the selection of one multipath route (set of next hops for a given destination) is referred to as 'policy'.
	
	Unless the mechanism indicated by ipForwardProto specifies otherwise, the policy specifier is the IP TOS Field. The encoding of IP TOS is as specified by the following convention. Zero indicates the default path if no more specific policy applies.
	
	`+----+----+----+----+----+----+----+----+`
	`|         |              |    |    |`
	`|  PRECEDENCE  |  TYPE OF SERVICE  |  0 |`
	`|         |              |    |    |`
	`+----+----+----+----+----+----+----+----+`
	
	IP TOS IP TOS
	Field Policy Field Policy
	Contents Code Contents Code
	0 0 0 0 ==> 0 0 0 0 1 ==> 2
	0 0 1 0 ==> 4 0 0 1 1 ==> 6
	0 1 0 0 ==> 8 0 1 0 1 ==> 10
	0 1 1 0 ==> 12 0 1 1 1 ==> 14
	1 0 0 0 ==> 16 1 0 0 1 ==> 18
	1 0 1 0 ==> 20 1 0 1 1 ==> 22
	1 1 0 0 ==> 24 1 1 0 1 ==> 26
	1 1 1 0 ==> 28 1 1 1 1 ==> 30
	
	Protocols defining 'policy' otherwise must either define a set of values which are valid for this object or must implement an integer-instanced policy table for which this object's value acts as an index.

OID	{ ipForwardEntry 3 }
NOTE	

OBJECT	ipForwardNextHop
SYNTAX	IpAddress
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	On remote routes, the address of the next sys-

	temen route; Otherwise, 0.0.0.0.
OID	{ ipForwardEntry 4 }
NOTE	

OBJECT	ipForwardIfIndex
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	The ifIndex value which identifies the local interface through which the next hop of this route should be reached.
OID	{ ipForwardEntry 5 }
NOTE	

OBJECT	ipForwardType
SYNTAX	INTEGER { other (1), invalid (2), local (3), remote (4) }
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	The type of route. Note that local(3) refers to a route for which the next hop is the final destination; remote(4) refers to a route for which the next hop is not the final destination.

Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipForwardTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said 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 ipForwardType object. |
| OID | { ipForwardEntry 6 } |
| NOTE | |

OBJECT	ipForwardProto
SYNTAX	INTEGER { other (1), local (2), netmgmt (3), icmp (4), egp (5), ggp (6), hello (7), rip (8), is-is (9), es-is (10), ciscoIgrp (11), bbnSpfIgp (12), ospf (13), bgp (14), idpr (15) }

MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.
OID	{ ipForwardEntry 7 }
NOTE	

OBJECT	ipForwardAge
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	obsolete
DSCRPT	The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of `too old' can be implied except through knowledge of the routing protocol by which the route was learned.
OID	{ ipForwardEntry 8 }
NOTE	

OBJECT	ipForwardInfo
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipForwardProto 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, and any implementation conforming to ASN.1 and the Basic Encoding Rules must be able to generate and recognize this value.
OID	{ ipForwardEntry 9 }
NOTE	

OBJECT	ipForwardNextHopAS
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	The Autonomous System Number of the Next Hop. When this is unknown or not relevant to the

	protocol indicated by ipForwardProto, zero.
OID	{ ipForwardEntry 10 }
NOTE	

OBJECT	ipForwardMetric1
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipForwardProto value. If this metric is not used, its value should be set to -1.
OID	{ ipForwardEntry 11 }
NOTE	

OBJECT	ipForwardMetric2
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipForwardProto value. If this metric is not used, its value should be set to -1.
OID	{ ipForwardEntry 12 }
NOTE	

OBJECT	ipForwardMetric3
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipForwardProto value. If this metric is not used, its value should be set to -1.
OID	{ ipForwardEntry 13 }
NOTE	

OBJECT	ipForwardMetric4
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete

DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipForwardProto value. If this metric is not used, its value should be set to -1.
OID	{ ipForwardEntry 14 }
NOTE	

OBJECT	ipForwardMetric5
SYNTAX	Integer32
MAX-ACCESS	read-create
STATUS	obsolete
DSCRPT	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipForwardProto value. If this metric is not used, its value should be set to -1.
OID	{ ipForwardEntry 15 }
NOTE	

COMPLIANCE	ipForwardOldCompliance
STATUS	obsolete
DSCRPT	The compliance statement for SNMP entities which implement the ipForward MIB.
MANDATORY-G	ipForwardMultiPathGroup
OID	{ ipForwardCompliances 2 }
NOTE	

OBJECT-G	ipForwardMultiPathGroup
OBJECTS	ipForwardNumber, ipForwardDest, ipForwardMask, ipForwardPolicy, ipForwardNextHop, ipForwardIfIndex, ipForwardType, ipForwardProto, ipForwardAge, ipForwardInfo, ipForwardNextHopAS, ipForwardMetric1, ipForwardMetric2, ipForwardMetric3, ipForwardMetric4, ipForwardMetric5
STATUS	obsolete
DSCRPT	IP Multipath Route Table.
OID	{ ipForwardGroups 2 }
NOTE	

3.10 IF-MIB

MODULE	ifMIB
UPDATED	200006140000Z
DSCRPT	The MIB module to describe generic objects for network

	interface sub-layers. This MIB is an updated version of MIB-II's ifTable, and incorporates the extensions defined in RFC 1229.
REVISION	200006140000Z
DSCRPT	Clarifications agreed upon by the Interfaces MIB WG, and published as RFC 2863.
REVISION	199602282155Z
DSCRPT	Revisions made by the Interfaces MIB WG, and published in RFC 2233.
REVISION	199311082155Z
DSCRPT	Initial revision, published as part of RFC 1573.
OID	{ mib-2 31 }
NOTE	

TC	OwnerString
STATUS	deprecated
DSCRPT	This data type is used to model an administratively assigned name of the owner of a resource. This information is taken from the NVT ASCII character set. It is suggested that this name contain one or more of the following: ASCII form of the manager station's transport address, management station name (e.g., domain name), network management personnel's name, location, or phone number. In some cases the agent itself will be the owner of an entry. In these cases, this string shall be set to a string starting with 'agent'.
SYNTAX	OCTET STRING (SIZE(0..255))

TC	InterfaceIndex
STATUS	current
DSCRPT	A unique value, greater than zero, for each interface or interface sub-layer in the managed system. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.
SYNTAX	Integer32 (1..2147483647)

TC	InterfaceIndexOrZero
STATUS	current
DSCRPT	This textual convention is an extension of the InterfaceIndex convention. The latter defines a greater than zero value used to identify an interface or interface sub-layer in the managed system. This extension permits the additional value of zero. the value zero is object-specific and must therefore be defined as part of the description of

	any object which uses this syntax. Examples of the usage of zero might include situations where interface was unknown, or when none or all interfaces need to be referenced.
SYNTAX	Integer32 (0..2147483647)

OBJECT	ifNumber
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of network interfaces (regardless of their current state) present on this system.
OID	{ interfaces 1 }
NOTE	

OBJECT	ifTableLastChange
SYNTAX	TimeTicks
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ ifMIBObjects 5 }
NOTE	

OBJECT	ifTable
SYNTAX	SEQUENCE OF IfEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A list of interface entries. The number of entries is given by the value of ifNumber.
OID	{ interfaces 2 }
NOTE	

OBJECT	ifEntry
SYNTAX	IfEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An entry containing management information applicable to a particular interface.
INDEX	ifIndex
OID	{ ifTable 1 }

NOTE	

OBJECT	ifIndex
SYNTAX	InterfaceIndex
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.
OID	{ ifEntry 1 }
NOTE	ifIndex Loopback 1, 2 LAN 3 EWAN 4, 5 PPPoE 6-10 Dialer 11-30 (FITELnet-F100) IPSec-IF 31-34 (FITELnet-F100) Null-IF 35 FITELnet-F100 11 FITELnet-F1000

OBJECT	ifDescr
SYNTAX	DisplayString (SIZE (0..255))
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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 interface hardware/software.
OID	{ ifEntry 2 }
NOTE	ifDescr Loopback softwareLoopback(24) LAN ethernetCsmacd(6) EWAN ethernetCsmacd(6) PPPoE ppp(23) Dialer basicISDN(20) IPSec-IF other(1) Null-IF other(1)

OBJECT	ifType
SYNTAX	IANAifType
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The type of interface. Additional values for ifType are

	assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.
OID	{ ifEntry 3 }
NOTE	ifName Loopback loopback 0, loopback 1 LAN lan 1 EWAN ewan 1, ewan 2 PPPoE pppoe 1 pppoe5 Dialer dialer 1 dialer 20 IPSec-IF ipsecif 1 ipsecif 4 Null-IF null 0

OBJECT	ifMtu
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The size of the largest packet which 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.
OID	{ ifEntry 4 }
NOTE	

OBJECT	ifSpeed
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	An estimate of the interface's current bandwidth in bits per second. For interfaces which 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 should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interace's speed. For a sub-layer which has no concept of bandwidth, this object should be zero.
OID	{ ifEntry 5 }
NOTE	

OBJECT	ifPhysAddress
SYNTAX	PhysAddress
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The interface's address at its protocol sub-layer. For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.
OID	{ ifEntry 6 }
NOTE	

OBJECT	ifAdminStatus
SYNTAX	INTEGER { up(1), down(2), testing(3) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up(1) or testing(3) states (or remains in the down(2) state).
OID	{ ifEntry 7 }
NOTE	

OBJECT	ifOperStatus
SYNTAX	INTEGER { up(1), down(2), testing(3), unknown(4), dormant(5), notPresent(6), lowerLayerDown(7) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up(1) then ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(6) state if the interface has missing (typically, hardware) components.
OID	{ ifEntry 8 }
NOTE	

OBJECT	ifLastChange
SYNTAX	TimeTicks
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ ifEntry 9 }
NOTE	

OBJECT	ifInOctets
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 10 }
NOTE	

OBJECT	ifInUcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 11 }
NOTE	

OBJECT	ifInNUcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	deprecated
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. This object is deprecated in favour of ifInMulticastPkts and ifInBroadcastPkts.
OID	{ ifEntry 12 }
NOTE	

OBJECT	ifInDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of inbound packets which 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. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 13 }
NOTE	

OBJECT	ifInErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of

	ifCounterDiscontinuityTime.
OID	{ ifEntry 14 }
NOTE	

OBJECT	ifInUnknownProtos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0.
	Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 15 }
NOTE	

OBJECT	ifOutOctets
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of octets transmitted out of the interface, including framing characters.
	Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 16 }
NOTE	

OBJECT	ifOutUcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a

	multicast or broadcast address at this sub-layer, including those that were discarded or not sent.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. |
| OID | { ifEntry 17 } |
| NOTE | |

OBJECT	ifOutNUcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	deprecated
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

This object is deprecated in favour of ifOutMulticastPkts and ifOutBroadcastPkts. |
| OID | { ifEntry 18 } |
| NOTE | |

OBJECT	ifOutDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of outbound packets which 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.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. |
| OID | { ifEntry 19 } |
| NOTE | |

OBJECT	ifOutErrors

SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifEntry 20 }
NOTE	

OBJECT	ifOutQLen
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	deprecated
DSCRPT	The length of the output packet queue (in packets).
OID	{ ifEntry 21 }
NOTE	

OBJECT	ifSpecific
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-only
STATUS	deprecated
DSCRPT	A reference to MIB definitions specific to the particular media being used to realize the interface. It is recommended that this value point to an instance of a MIB object in the media-specific MIB, i.e., that this object have the semantics associated with the InstancePointer textual convention defined in RFC 2579. In fact, it is recommended that the media-specific MIB specify what value ifSpecific should/can take for values of ifType. If no MIB definitions specific to the particular media are available, the value should be set to the OBJECT IDENTIFIER { 0 0 }.
OID	{ ifEntry 22 }
NOTE	

OBJECT	ifXTable
SYNTAX	SEQUENCE OF IfXEntry
MAX-ACCESS	not-accessible
STATUS	current

DSCRPT	A list of interface entries. The number of entries is given by the value of ifNumber. This table contains additional objects for the interface table.
OID	{ ifMIBObjects 1 }
NOTE	

OBJECT	ifXEntry
SYNTAX	IfXEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An entry containing additional management information applicable to a particular interface.
AUGMENTS	ifEntry
OID	{ ifXTable 1 }
NOTE	

OBJECT	ifName
SYNTAX	DisplayString
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's `console'.  This might be a text name, such as `le0' or a simple port number, such as `1', depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. Note that for an agent which responds to SNMP queries concerning an interface on some other (proxied) device, then the value of ifName for such an interface is the proxied device's local name for it.

If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string. |
| OID | { ifXEntry 1 } |
| NOTE | |

OBJECT	ifInMulticastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this

	includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 2 }
NOTE	

OBJECT	ifInBroadcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 3 }
NOTE	

OBJECT	ifOutMulticastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which 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. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 4 }
NOTE	

OBJECT	ifOutBroadcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 5 }
NOTE	

OBJECT	ifHCInOctets
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 6 }
NOTE	

OBJECT	ifHCInUcastPkts
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. This object is a 64-bit version of ifInUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 7 }
NOTE	

OBJECT	ifHCInMulticastPkts
SYNTAX	Counter64

MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 8 }
NOTE	

OBJECT	ifHCInBroadcastPkts
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. This object is a 64-bit version of ifInBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 9 }
NOTE	

OBJECT	ifHCOutOctets
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 10 }
NOTE	

OBJECT	ifHCOutUcastPkts
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 11 }
NOTE	

OBJECT	ifHCOutMulticastPkts
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which 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. This object is a 64-bit version of ifOutMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 12 }
NOTE	

OBJECT	ifHCOutBroadcastPkts
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutBroadcastPkts. Discontinuities in the value of this counter can occur at

	re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
OID	{ ifXEntry 13 }
NOTE	

OBJECT	ifLinkUpDownTrapEnable
SYNTAX	INTEGER { enabled(1), disabled(2) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	Indicates whether linkUp/linkDown traps should be generated for this interface. By default, this object should have the value enabled(1) for interfaces which do not operate on 'top' of any other interface (as defined in the ifStackTable), and disabled(2) otherwise.
OID	{ ifXEntry 14 }
NOTE	

OBJECT	ifHighSpeed
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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 which 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 which has no concept of bandwidth, this object should be zero.
OID	{ ifXEntry 15 }
NOTE	

OBJECT	ifPromiscuousMode
SYNTAX	TruthValue
MAX-ACCESS	read-write
STATUS	current
DSCRPT	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. The value true(1) is only legal on certain types of media. If legal, setting this object to a value of true(1) may require the

	interface to be reset before becoming effective. The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface.
OID	{ ifXEntry 16 }
NOTE	

OBJECT	ifConnectorPresent
SYNTAX	TruthValue
MAX-ACCESS	read-only
STATUS	current
DSCRPT	This object has the value 'true(1)' if the interface sublayer has a physical connector and the value 'false(2)' otherwise.
OID	{ ifXEntry 17 }
NOTE	

OBJECT	ifAlias
SYNTAX	DisplayString (SIZE(0..64))
MAX-ACCESS	read-write
STATUS	current
DSCRPT	This object is an 'alias' name for the interface as specified by a network manager, and provides a non-volatile 'handle' for the interface. On the first instantiation of an interface, the value of ifAlias associated with that interface is the zero-length string. As and when a value is written into an instance of ifAlias through a network management set operation, then the agent must retain the supplied value in the ifAlias instance associated with the same interface for as long as that interface remains instantiated, including across all re-initializations/reboots of the network management system, including those which result in a change of the interface's ifIndex value. An example of the value which a network manager might store in this object for a WAN interface is the (Telco's) circuit number/identifier of the interface. Some agents may support write-access only for interfaces having particular values of ifType. An agent which supports write access to this object is required to keep the value in non-volatile storage, but it may limit the length of new values depending on how much storage is already occupied by the current values for other interfaces.

OID	{ ifXEntry 18 }
NOTE	

OBJECT	ifCounterDiscontinuityTime
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime on the most recent occasion at which 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.
OID	{ ifXEntry 19 }
NOTE	

OBJECT	ifStackTable
SYNTAX	SEQUENCE OF IfStackEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The table containing information on the relationships between the multiple sub-layers of network interfaces. In particular, it contains information on which sub-layers run 'on top of' which other sub-layers, where each sub-layer corresponds to a conceptual row in the ifTable. For example, when the sub-layer with ifIndex value x runs over the sub-layer with ifIndex value y, then this table contains: ifStackStatus.x.y=active For each ifIndex value, I, which identifies an active interface, there are always at least two instantiated rows in this table associated with I. For one of these rows, I is the value of ifStackHigherLayer; for the other, I is the value of ifStackLowerLayer. (If I is not involved in multiplexing, then these are the only two rows associated with I.) For example, two rows exist even for an interface which has no others stacked on top or below it: ifStackStatus.0.x=active ifStackStatus.x.0=active
OID	{ ifMIBObjects 2 }

NOTE	

OBJECT	ifStackEntry
SYNTAX	IfStackEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	Information on a particular relationship between two sub-layers, specifying that one sub-layer runs on 'top' of the other sub-layer. Each sub-layer corresponds to a conceptual row in the ifTable.
INDEX	ifStackHigherLayer, ifStackLowerLayer
OID	{ ifStackTable 1 }
NOTE	

OBJECT	ifStackHigherLayer
SYNTAX	InterfaceIndexOrZero
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The value of ifIndex corresponding to the higher sub-layer of the relationship, i.e., the sub-layer which runs on 'top' of the sub-layer identified by the corresponding instance of ifStackLowerLayer. If there is no higher sub-layer (below the internetwork layer), then this object has the value 0.
OID	{ ifStackEntry 1 }
NOTE	

OBJECT	ifStackLowerLayer
SYNTAX	InterfaceIndexOrZero
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The value of ifIndex corresponding to the lower sub-layer of the relationship, i.e., the sub-layer which runs 'below' the sub-layer identified by the corresponding instance of ifStackHigherLayer. If there is no lower sub-layer, then this object has the value 0.
OID	{ ifStackEntry 2 }
NOTE	

OBJECT	ifStackStatus
SYNTAX	RowStatus
MAX-ACCESS	read-create
STATUS	current
DSCRPT	The status of the relationship between two sub-layers.

	Changing the value of this object from 'active' to 'notInService' or 'destroy' will likely have consequences up and down the interface stack. Thus, write access to this object is likely to be inappropriate for some types of interfaces, and many implementations will choose not to support write-access for any type of interface.
OID	{ ifStackEntry 3 }
NOTE	

OBJECT	ifStackLastChange
SYNTAX	TimeTicks
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime at the time of the last change of the (whole) interface stack. A change of the interface stack is defined to be any creation, deletion, or change in value of any instance of ifStackStatus. If the interface stack has been unchanged since the last re-initialization of the local network management subsystem, then this object contains a zero value.
OID	{ ifMIBObjects 6 }
NOTE	

OBJECT	ifRcvAddressTable
SYNTAX	SEQUENCE OF IfRcvAddressEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	This table contains an entry for each address (broadcast, multicast, or uni-cast) for which the system will receive packets/frames on a particular interface, except as follows: - for an interface operating in promiscuous mode, entries are only required for those addresses for which the system would receive frames were it not operating in promiscuous mode. - for 802.5 functional addresses, only one entry is required, for the address which has the functional address bit ANDed with the bit mask of all functional addresses for which the interface will accept frames. A system is normally able to use any unicast address which corresponds to an entry in this table as a source address.
OID	{ ifMIBObjects 4 }
NOTE	

OBJECT	ifRcvAddressEntry

SYNTAX	IfRcvAddressEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A list of objects identifying an address for which the system will accept packets/frames on the particular interface identified by the index value ifIndex.
INDEX	ifIndex, ifRcvAddressAddress
OID	{ ifRcvAddressTable 1 }
NOTE	

OBJECT	ifRcvAddressAddress
SYNTAX	PhysAddress
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An address for which the system will accept packets/frames on this entry's interface.
OID	{ ifRcvAddressEntry 1 }
NOTE	

OBJECT	ifRcvAddressStatus
SYNTAX	RowStatus
MAX-ACCESS	read-create
STATUS	current
DSCRPT	This object is used to create and delete rows in the ifRcvAddressTable.
OID	{ ifRcvAddressEntry 2 }
NOTE	

OBJECT	ifRcvAddressType
SYNTAX	INTEGER { other(1), volatile(2), nonVolatile(3) }
MAX-ACCESS	read-create
STATUS	current
DSCRPT	This object has the value nonVolatile(3) for those entries in the table which are valid and will not be deleted by the next restart of the managed system. Entries having the value volatile(2) are valid and exist, but have not been saved, so that will not exist after the next restart of the managed system. Entries having the value other(1) are valid and exist but are not classified as to whether they will continue to exist after the next restart.
OID	{ ifRcvAddressEntry 3 }
NOTE	

TRAP	linkDown
OBJECTS	ifIndex, ifAdminStatus, ifOperStatus
STATUS	current
DSCRPT	A linkDown trap signifies that the SNMP entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links is about to enter the down state from some other state (but not from the notPresent state). This other state is indicated by the included value of ifOperStatus.
OID	{ snmpTraps 3 }
NOTE	

TRAP	linkUp
OBJECTS	ifIndex, ifAdminStatus, ifOperStatus
STATUS	current
DSCRPT	A linkUp trap signifies that the SNMP entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links left the down state and transitioned into some other state (but not into the notPresent state). This other state is indicated by the included value of ifOperStatus.
OID	{ snmpTraps 4 }
NOTE	

COMPLIANCE	ifCompliance3
STATUS	current
DSCRPT	The compliance statement for SNMP entities which have network interfaces.
MANDATORY-G	ifGeneralInformationGroup, linkUpDownNotificationsGroup
GROUP	ifFixedLengthGroup
DSCRPT	This group is mandatory for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is less than or equal to 20,000,000 bits/second.
GROUP	ifHCFixedLengthGroup
DSCRPT	This group is mandatory for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.
GROUP	ifPacketGroup
DSCRPT	This group is mandatory for those network interfaces which are packet-oriented, and for which the value of the corresponding instance of ifSpeed is less than or equal to

	20,000,000 bits/second.
GROUP	ifHCPacketGroup
DSCRPT	This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second but less than or equal to 650,000,000 bits/second.
GROUP	ifVHCPacketGroup
DSCRPT	This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.
GROUP	ifCounterDiscontinuityGroup
DSCRPT	This group is mandatory for those network interfaces that are required to maintain counters (i.e., those for which one of the ifFixedLengthGroup, ifHCFixedLengthGroup, ifPacketGroup, ifHCPacketGroup, or ifVHCPacketGroup is mandatory).
GROUP	ifRcvAddressGroup
DSCRPT	The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.
OBJECT	ifLinkUpDownTrapEnable
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifPromiscuousMode
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifAdminStatus
SYNTAX	INTEGER { up(1), down(2) }
MIN-ACCESS	read-only
DSCRPT	Write access is not required, nor is support for the value testing(3).
OBJECT	ifAlias
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OID	{ ifCompliances 3 }
NOTE	

OBJECT-G	ifGeneralInformationGroup
OBJECTS	ifIndex, ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifLinkUpDownTrapEnable, ifConnectorPresent, ifHighSpeed, ifName, ifNumber, ifAlias, ifTableLastChange
STATUS	current

DSCRPT	A collection of objects providing information applicable to all network interfaces.
CID	{ ifGroups 10 }
NOTE	

OBJECT-G	ifFixedLengthGroup
OBJECTS	ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors
STATUS	current
DSCRPT	A collection of objects providing information specific to non-high speed (non-high speed interfaces transmit and receive at speeds less than or equal to 20,000,000 bits/second) character-oriented or fixed-length-transmission network interfaces.
CID	{ ifGroups 2 }
NOTE	

OBJECT-G	ifHCFixedLengthGroup
OBJECTS	ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors
STATUS	current
DSCRPT	A collection of objects providing information specific to high speed (greater than 20,000,000 bits/second) character-oriented or fixed-length-transmission network interfaces.
CID	{ ifGroups 3 }
NOTE	

OBJECT-G	ifPacketGroup
OBJECTS	ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode
STATUS	current
DSCRPT	A collection of objects providing information specific to non-high speed (non-high speed interfaces transmit and receive at speeds less than or equal to 20,000,000 bits/second) packet-oriented network interfaces.
CID	{ ifGroups 4 }
NOTE	

OBJECT-G	ifHCPacketGroup
OBJECTS	ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode

STATUS	current
DSCRPT	A collection of objects providing information specific to high speed (greater than 20,000,000 bits/second but less than or equal to 650,000,000 bits/second) packet-oriented network interfaces.
OID	{ ifGroups 5 }
NOTE	

OBJECT-G	ifVHCPacketGroup
OBJECTS	ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOutUcastPkts, ifHCOutMulticastPkts, ifHCOutBroadcastPkts, ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode
STATUS	current
DSCRPT	A collection of objects providing information specific to higher speed (greater than 650,000,000 bits/second) packet-oriented network interfaces.
OID	{ ifGroups 6 }
NOTE	

OBJECT-G	ifRcvAddressGroup
OBJECTS	ifRcvAddressStatus, ifRcvAddressType
STATUS	current
DSCRPT	A collection of objects providing information on the multiple addresses which an interface receives.
OID	{ ifGroups 7 }
NOTE	

OBJECT-G	ifStackGroup2
OBJECTS	ifStackStatus, ifStackLastChange
STATUS	current
DSCRPT	A collection of objects providing information on the layering of MIB-II interfaces.
OID	{ ifGroups 11 }
NOTE	

OBJECT-G	ifCounterDiscontinuityGroup
OBJECTS	ifCounterDiscontinuityTime
STATUS	current
DSCRPT	A collection of objects providing information specific to interface counter discontinuities.
OID	{ ifGroups 13 }

NOTE	

NOTIFICATION-G	linkUpDownNotificationsGroup
NOTIFICATIONS	linkUp, linkDown
STATUS	current
DSCRPT	The notifications which indicate specific changes in the value of ifOperStatus.
OID	{ ifGroups 14 }
NOTE	

OBJECT	ifTestTable
SYNTAX	SEQUENCE OF IfTestEntry
MAX-ACCESS	not-accessible
STATUS	deprecated

DSCRPT	This table contains one entry per interface. It defines objects which allow a network manager to instruct an agent to test an interface for various faults. Tests for an interface are defined in the media-specific MIB for that interface. After invoking a test, the object ifTestResult can be read to determine the outcome. If an agent can not perform the test, ifTestResult is set to so indicate. The object ifTestCode can be used to provide further test-specific or interface-specific (or even enterprise-specific) information concerning the outcome of the test. Only one test can be in progress on each interface at any one time. If one test is in progress when another test is invoked, the second test is rejected. Some agents may reject a test when a prior test is active on another interface.
	
	Before starting a test, a manager-station must first obtain 'ownership' of the entry in the ifTestTable for the interface to be tested. This is accomplished with the ifTestId and ifTestStatus objects as follows:
	
	try_again:
	get (ifTestId, ifTestStatus)
	while (ifTestStatus != notInUse)
	/*
	* Loop while a test is running or some other
	* manager is configuring a test.
	*/
	short delay
	get (ifTestId, ifTestStatus)
	}
	
	/*
	* Is not being used right now -- let's compete

```
* to see who gets it.
*/
lock_value = ifTestId

if ( set(ifTestId = lock_value, ifTestStatus = inUse,
ifTestOwner = 'my-IP-address') == FAILURE)
/*
* Another manager got the ifTestEntry -- go
* try again
*/
goto try_again;

/*
* I have the lock
*/
set up any test parameters.

/*
* This starts the test
*/
set(ifTestType = test_to_run);

wait for test completion by polling ifTestResult

when test completes, agent sets ifTestResult
agent also sets ifTestStatus = 'notInUse'

retrieve any additional test results, and ifTestId

if (ifTestId == lock_value+1) results are valid
```

A manager station first retrieves the value of the
appropriate ifTestId and ifTestStatus objects, periodically
repeating the retrieval if necessary, until the value of
ifTestStatus is 'notInUse'. The manager station then tries
to set the same ifTestId object to the value it just
retrieved, the same ifTestStatus object to 'inUse', and the
corresponding ifTestOwner object to a value indicating
itself. If the set operation succeeds then the manager has
obtained ownership of the ifTestEntry, and the value of the
ifTestId object is incremented by the agent (per the
semantics of TestAndIncr). Failure of the set operation
indicates that some other manager has obtained ownership of
the ifTestEntry.

Once ownership is obtained, any test parameters can be
setup, and then the test is initiated by setting ifTestType.
On completion of the test, the agent sets ifTestStatus to
'notInUse'. Once this occurs, the manager can retrieve the
results. In the (rare) event that the invocation of tests

	by two network managers were to overlap, then there would be a possibility that the first test's results might be overwritten by the second test's results prior to the first results being read. This unlikely circumstance can be detected by a network manager retrieving ifTestId at the same time as retrieving the test results, and ensuring that the results are for the desired request. If ifTestType is not set within an abnormally long period of time after ownership is obtained, the agent should time-out the manager, and reset the value of the ifTestStatus object back to 'notInUse'. It is suggested that this time-out period be 5 minutes. In general, a management station must not retransmit a request to invoke a test for which it does not receive a response; instead, it properly inspects an agent's MIB to determine if the invocation was successful. Only if the invocation was unsuccessful, is the invocation request retransmitted. Some tests may require the interface to be taken off-line in order to execute them, or may even require the agent to reboot after completion of the test. In these circumstances, communication with the management station invoking the test may be lost until after completion of the test. An agent is not required to support such tests. However, if such tests are supported, then the agent should make every effort to transmit a response to the request which invoked the test prior to losing communication. When the agent is restored to normal service, the results of the test are properly made available in the appropriate objects. Note that this requires that the ifIndex value assigned to an interface must be unchanged even if the test causes a reboot. An agent must reject any test for which it cannot, perhaps due to resource constraints, make available at least the minimum amount of information after that test completes.
OID	{ ifMIBObjects 3 }
NOTE	

OBJECT	ifTestEntry
SYNTAX	IfTestEntry
MAX-ACCESS	not-accessible
STATUS	deprecated
DSCRPT	An entry containing objects for invoking tests on an interface.
AUGMENTS	ifEntry

OID	{ ifTestTable 1 }
NOTE	

OBJECT	ifTestId
SYNTAX	TestAndIncr
MAX-ACCESS	read-write
STATUS	deprecated
DSCRPT	This object identifies the current invocation of the interface's test.
OID	{ ifTestEntry 1 }
NOTE	

OBJECT	ifTestStatus
SYNTAX	INTEGER { notInUse(1), inUse(2) }
MAX-ACCESS	read-write
STATUS	deprecated
DSCRPT	This object indicates whether or not some manager currently has the necessary 'ownership' required to invoke a test on this interface. A write to this object is only successful when it changes its value from 'notInUse(1)' to 'inUse(2)'. After completion of a test, the agent resets the value back to 'notInUse(1)'.
OID	{ ifTestEntry 2 }
NOTE	

OBJECT	ifTestType
SYNTAX	AutonomousType
MAX-ACCESS	read-write
STATUS	deprecated
DSCRPT	A control variable used to start and stop operator-initiated interface tests. Most OBJECT IDENTIFIER values assigned to tests are defined elsewhere, in association with specific types of interface. However, this document assigns a value for a full-duplex loopback test, and defines the special meanings of the subject identifier: noTest OBJECT IDENTIFIER ::= { 0 0 } When the value noTest is written to this object, no action is taken unless a test is in progress, in which case the test is aborted. Writing any other value to this object is only valid when no test is currently in progress, in which case the indicated test is initiated. When read, this object always returns the most recent value

	that ifTestType was set to. If it has not been set since the last initialization of the network management subsystem on the agent, a value of noTest is returned.
OID	{ ifTestEntry 3 }
NOTE	

OBJECT	ifTestResult
SYNTAX	INTEGER { none(1), success(2), inProgress(3), notSupported(4), unAbleToRun(5), aborted(6), failed(7) }
MAX-ACCESS	read-only
STATUS	deprecated
DSCRPT	This object contains the result of the most recently requested test, or the value none(1) if no tests have been requested since the last reset. Note that this facility provides no provision for saving the results of one test when starting another, as could be required if used by multiple managers concurrently.
OID	{ ifTestEntry 4 }
NOTE	

OBJECT	ifTestCode
SYNTAX	OBJECT IDENTIFIER
MAX-ACCESS	read-only
STATUS	deprecated
DSCRPT	This object contains a code which contains more specific information on the test result, for example an error-code after a failed test. Error codes and other values this object may take are specific to the type of interface and/or test. The value may have the semantics of either the AutonomousType or InstancePointer textual conventions as defined in RFC 2579. The identifier: testCodeUnknown OBJECT IDENTIFIER ::= { 0 0 } is defined for use if no additional result code is available.
OID	{ ifTestEntry 5 }
NOTE	

OBJECT	ifTestOwner
SYNTAX	OwnerString
MAX-ACCESS	read-write
STATUS	deprecated
DSCRPT	The entity which currently has the 'ownership' required to invoke a test on this interface.

OID	{ ifTestEntry 6 }
NOTE	

OBJECT-G	ifGeneralGroup
OBJECTS	ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifLinkUpDownTrapEnable, ifConnectorPresent, ifHighSpeed, ifName
STATUS	deprecated
DSCRPT	A collection of objects deprecated in favour of ifGeneralInformationGroup.
OID	{ ifGroups 1 }
NOTE	

OBJECT-G	ifTestGroup
OBJECTS	ifTestId, ifTestStatus, ifTestType, ifTestResult, ifTestCode, ifTestOwner
STATUS	deprecated
DSCRPT	A collection of objects providing the ability to invoke tests on an interface.
OID	{ ifGroups 8 }
NOTE	

OBJECT-G	ifStackGroup
OBJECTS	ifStackStatus
STATUS	deprecated
DSCRPT	The previous collection of objects providing information on the layering of MIB-II interfaces.
OID	{ ifGroups 9 }
NOTE	

OBJECT-G	ifOldObjectsGroup
OBJECTS	ifInNUcastPkts, ifOutNUcastPkts, ifOutQLen, ifSpecific
STATUS	deprecated
DSCRPT	The collection of objects deprecated from the original MIB-II interfaces group.
OID	{ ifGroups 12 }
NOTE	

COMPLIANCE	ifCompliance
STATUS	deprecated
DSCRPT	A compliance statement defined in a previous version of this MIB module, for SNMP entities which have network interfaces.

MANDATORY-G	ifGeneralGroup, ifStackGroup
GROUP	ifFixedLengthGroup
DSCRPT	This group is mandatory for all network interfaces which are character-oriented or transmit data in fixed-length transmission units.
GROUP	ifHCFixedLengthGroup
DSCRPT	This group is mandatory only for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.
GROUP	ifPacketGroup
DSCRPT	This group is mandatory for all network interfaces which are packet-oriented.
GROUP	ifHCPacketGroup
DSCRPT	This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.
GROUP	ifTestGroup
DSCRPT	This group is optional. Media-specific MIBs which require interface tests are strongly encouraged to use this group for invoking tests and reporting results. A medium specific MIB which has mandatory tests may make implementation of this group mandatory.
GROUP	ifRcvAddressGroup
DSCRPT	The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.
OBJECT	ifLinkUpDownTrapEnable
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifPromiscuousMode
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifStackStatus
SYNTAX	INTEGER { active(1) }
MIN-ACCESS	read-only
DSCRPT	Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1).
OBJECT	ifAdminStatus
SYNTAX	INTEGER { up(1), down(2) }
MIN-ACCESS	read-only

DSCRPT	Write access is not required, nor is support for the value testing(3).
OID	{ ifCompliances 1 }
NOTE	

COMPLIANCE	ifCompliance2
STATUS	deprecated
DSCRPT	A compliance statement defined in a previous version of this MIB module, for SNMP entities which have network interfaces.
MANDATORY-G	ifGeneralInformationGroup, ifStackGroup2, ifCounterDiscontinuityGroup
GROUP	ifFixedLengthGroup
DSCRPT	This group is mandatory for all network interfaces which are character-oriented or transmit data in fixed-length transmission units.
GROUP	ifHCFixedLengthGroup
DSCRPT	This group is mandatory only for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.
GROUP	ifPacketGroup
DSCRPT	This group is mandatory for all network interfaces which are packet-oriented.
GROUP	ifHCPacketGroup
DSCRPT	This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.
GROUP	ifRcvAddressGroup
DSCRPT	The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.
OBJECT	ifLinkUpDownTrapEnable
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifPromiscuousMode
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OBJECT	ifStackStatus
SYNTAX	INTEGER { active(1) }
MIN-ACCESS	read-only
DSCRPT	Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need

	be supported, specifically: active(1).
OBJECT	ifAdminStatus
SYNTAX	INTEGER { up(1), down(2) }
MIN-ACCESS	read-only
DSCRPT	Write access is not required, nor is support for the value testing(3).
OBJECT	ifAlias
MIN-ACCESS	read-only
DSCRPT	Write access is not required.
OID	{ ifCompliances 2 }
NOTE	

3.11 IPV6-TC

TC	Ipv6Address
STATUS	current
DSCRPT	This data type is used to model IPv6 addresses. This is a binary string of 16 octets in network byte-order.
SYNTAX	OCTET STRING (SIZE (16))

TC	Ipv6AddressPrefix
STATUS	current
DSCRPT	This data type is used to model IPv6 address prefixes. This is a binary string of up to 16 octets in network byte-order.
SYNTAX	OCTET STRING (SIZE (0..16))

TC	Ipv6AddressIfIdentifier
STATUS	current
DSCRPT	This data type is used to model IPv6 address interface identifiers. This is a binary string of up to 8 octets in network byte-order.
SYNTAX	OCTET STRING (SIZE (0..8))

TC	Ipv6IfIndex
STATUS	current
DSCRPT	A unique value, greater than zero for each internetwork-layer interface in the managed system. It is recommended that values are assigned contiguously starting from 1. The value for each internetwork-layer interface must remain constant at least from one re-initialization of the entity's network management system to the next

	re-initialization.
SYNTAX	Integer32 (1..2147483647)

TC	Ipv6IfIndexOrZero
STATUS	current
DSCRPT	This textual convention is an extension of the Ipv6IfIndex convention. The latter defines a greater than zero value used to identify an IPv6 interface in the managed system. This extension permits the additional value of zero. The value zero is object-specific and must therefore be defined as part of the description of any object which uses this syntax. Examples of the usage of zero might include situations where interface was unknown, or when none or all interfaces need to be referenced.
SYNTAX	Integer32 (0..2147483647)

3.12 IPV6-MIB

MODULE	ipv6MIB
UPDATED	9802052155Z
DSCRPT	The MIB module for entities implementing the IPv6 protocol.
OID	{ mib-2 55 }
NOTE	

OBJECT	ipv6Forwarding
SYNTAX	INTEGER { forwarding(1), notForwarding(2) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The indication of whether this entity is acting as an IPv6 router in respect to the forwarding of datagrams received by, but not addressed to, this entity. IPv6 routers forward datagrams. IPv6 hosts do not (except those source-routed via the host). Note that 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 `wrongValue' response if a management station attempts to change this object to an inappropriate value.
OID	{ ipv6MIBObjects 1 }

NOTE	

OBJECT	ipv6DefaultHopLimit
SYNTAX	INTEGER(0..255)
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The default value inserted into the Hop Limit field of the IPv6 header of datagrams originated at this entity, whenever a Hop Limit value is not supplied by the transport layer protocol.
OID	{ ipv6MIBObjects 2 }
NOTE	

OBJECT	ipv6Interfaces
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IPv6 interfaces (regardless of their current state) present on this system.
OID	{ ipv6MIBObjects 3 }
NOTE	

OBJECT	ipv6IfTableLastChange
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime at the time of the last insertion or removal of an entry in the ipv6IfTable. 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.
OID	{ ipv6MIBObjects 4 }
NOTE	

OBJECT	ipv6IfTable
SYNTAX	SEQUENCE OF Ipv6IfEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The IPv6 Interfaces table contains information on the entity's internetwork-layer interfaces. An IPv6 interface constitutes a logical network layer attachment to the layer immediately below IPv6 including internet layer 'tunnels', such as

	tunnels over IPv4 or IPv6 itself.
OID	{ ipv6MIBObjects 5 }
NOTE	

OBJECT	ipv6IfEntry
SYNTAX	Ipv6IfEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An interface entry containing objects about a particular IPv6 interface.
INDEX	ipv6IfIndex
OID	{ ipv6IfTable 1 }
NOTE	

OBJECT	ipv6IfIndex
SYNTAX	Ipv6IfIndex
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A unique non-zero value identifying the particular IPv6 interface.
OID	{ ipv6IfEntry 1 }
NOTE	

OBJECT	ipv6IfDescr
SYNTAX	DisplayString
MAX-ACCESS	read-write
STATUS	current
DSCRPT	A textual string containing information about the interface. This string may be set by the network management system.
OID	{ ipv6IfEntry 2 }
NOTE	

OBJECT	ipv6IfLowerLayer
SYNTAX	VariablePointer
MAX-ACCESS	read-only
STATUS	current
DSCRPT	This object identifies the protocol layer over which this network interface operates. If this network interface operates over the data-link layer, then the value of this object refers to an instance of ifIndex [6]. If this network interface operates over an IPv4 interface, the value of this

	object refers to an instance of ipAdEntAddr [3]. If this network interface operates over another IPv6 interface, the value of this object refers to an instance of ipv6IfIndex. If this network interface is not currently operating over an active protocol layer, then the value of this object should be set to the OBJECT ID { 0 0 }.
OID	{ ipv6IfEntry 3 }
NOTE	

OBJECT	ipv6IfEffectiveMtu
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The size of the largest IPv6 packet which can be sent/received on the interface, specified in octets.
OID	{ ipv6IfEntry 4 }
NOTE	

OBJECT	ipv6IfReasmMaxSize
SYNTAX	Unsigned32 (0..65535)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The size of the largest IPv6 datagram which this entity can re-assemble from incoming IPv6 fragmented datagrams received on this interface.
OID	{ ipv6IfEntry 5 }
NOTE	

OBJECT	ipv6IfIdentifier
SYNTAX	Ipv6AddressIfIdentifier
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The Interface Identifier for this interface that is (at least) unique on the link this interface is attached to. The Interface Identifier is combined with an address prefix to form an interface address. By default, the Interface Identifier is autoconfigured according to the rules of the link type this interface is attached to.
OID	{ ipv6IfEntry 6 }

NOTE	

OBJECT	ipv6IfIdentifierLength
SYNTAX	INTEGER (0..64)
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The length of the Interface Identifier in bits.
OID	{ ipv6IfEntry 7 }
NOTE	

OBJECT	ipv6IfPhysicalAddress
SYNTAX	PhysAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The interface's physical address. For example, for an IPv6 interface attached to an 802.x link, this object normally contains a MAC address. Note that in some cases this address may differ from the address of the interface's protocol sub-layer. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.
OID	{ ipv6IfEntry 8 }
NOTE	

OBJECT	ipv6IfAdminStatus
SYNTAX	INTEGER { up(1), down(2) }
MAX-ACCESS	read-write
STATUS	current
DSCRPT	The desired state of the interface. When a managed system initializes, all IPv6 interfaces start with ipv6IfAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed system, ipv6IfAdminStatus is then changed to the up(1) state (or remains in the down(2) state).
OID	{ ipv6IfEntry 9 }
NOTE	

OBJECT	ipv6IfOperStatus
SYNTAX	INTEGER { up(1), down(2), noIfIdentifier(3), unknown(4), notPresent(5) }
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The current operational state of the interface. The noIfIdentifier(3) state indicates that no valid Interface Identifier is assigned to the interface. This state usually indicates that the link-local interface address failed Duplicate Address Detection. If ipv6IfAdminStatus is down(2) then ipv6IfOperStatus should be down(2). If ipv6IfAdminStatus is changed to up(1) then ipv6IfOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should remain in the down(2) or noIfIdentifier(3) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(5) state if the interface has missing (typically, lower layer) components.
OID	{ ipv6IfEntry 10 }
NOTE	

OBJECT	ipv6IfLastChange
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	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.
OID	{ ipv6IfEntry 11 }
NOTE	

OBJECT	ipv6IfStatsTable
SYNTAX	SEQUENCE OF Ipv6IfStatsEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	IPv6 interface traffic statistics.
OID	{ ipv6MIBObjects 6 }
NOTE	

OBJECT	ipv6IfStatsEntry
SYNTAX	Ipv6IfStatsEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An interface statistics entry containing objects

	at a particular IPv6 interface.
AUGMENTS	ipv6IfEntry
OID	{ ipv6IfStatsTable 1 }
NOTE	

OBJECT	ipv6IfStatsInReceives
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of input datagrams received by the interface, including those received in error.
OID	{ ipv6IfStatsEntry 1 }
NOTE	

OBJECT	ipv6IfStatsInHdrErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams discarded due to errors in their IPv6 headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IPv6 options, etc.
OID	{ ipv6IfStatsEntry 2 }
NOTE	

OBJECT	ipv6IfStatsInTooBigErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams that could not be forwarded because their size exceeded the link MTU of outgoing interface.
OID	{ ipv6IfStatsEntry 3 }
NOTE	

OBJECT	ipv6IfStatsInNoRoutes
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams discarded because no route could be found to transmit them to their destination.

OID	{ ipv6IfStatsEntry 4 }
NOTE	

OBJECT	ipv6IfStatsInAddrErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams discarded because the IPv6 address in their IPv6 header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., ::0) and unsupported addresses (e.g., addresses with unallocated prefixes). For entities which are not IPv6 routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
OID	{ ipv6IfStatsEntry 5 }
NOTE	

OBJECT	ipv6IfStatsInUnknownProtos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol. This counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the datagrams.
OID	{ ipv6IfStatsEntry 6 }
NOTE	

OBJECT	ipv6IfStatsInTruncatedPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input datagrams discarded because datagram frame didn't carry enough data.
OID	{ ipv6IfStatsEntry 7 }
NOTE	

OBJECT	ipv6IfStatsInDiscards

SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of input IPv6 datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.
OID	{ ipv6IfStatsEntry 8 }
NOTE	

OBJECT	ipv6IfStatsInDelivers
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of datagrams successfully delivered to IPv6 user-protocols (including ICMP). This counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the datagrams.
OID	{ ipv6IfStatsEntry 9 }
NOTE	

OBJECT	ipv6IfStatsOutForwDatagrams
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of output datagrams which this entity received and forwarded to their final destinations. In entities which do not act as IPv6 routers, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route processing was successful. Note that for a successfully forwarded datagram the counter of the outgoing interface is incremented.
OID	{ ipv6IfStatsEntry 10 }
NOTE	

OBJECT	ipv6IfStatsOutRequests
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The total number of IPv6 datagrams which local IPv6 user-protocols (including ICMP) supplied to IPv6 in requests for transmission. Note that this counter does not include any datagrams counted in ipv6IfStatsOutForwDatagrams.
OID	{ ipv6IfStatsEntry 11 }
NOTE	

OBJECT	ipv6IfStatsOutDiscards
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of output IPv6 datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipv6IfStatsOutForwDatagrams if any such packets met this (discretionary) discard criterion.
OID	{ ipv6IfStatsEntry 12 }
NOTE	

OBJECT	ipv6IfStatsOutFragOKs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IPv6 datagrams that have been successfully fragmented at this output interface.
OID	{ ipv6IfStatsEntry 13 }
NOTE	

OBJECT	ipv6IfStatsOutFragFails
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IPv6 datagrams that have been discarded because they needed to be fragmented at this output interface but could not be.
OID	{ ipv6IfStatsEntry 14 }
NOTE	

OBJECT	ipv6IfStatsOutFragCreates
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The number of output datagram fragments that have been generated as a result of fragmentation at this output interface.
OID	{ ipv6IfStatsEntry 15 }
NOTE	

OBJECT	ipv6IfStatsReasmReqds
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IPv6 fragments received which needed to be reassembled at this interface. Note that this counter is incremented at the interface to which these fragments were addressed which might not be necessarily the input interface for some of the fragments.
OID	{ ipv6IfStatsEntry 16 }
NOTE	

OBJECT	ipv6IfStatsReasmOKs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of IPv6 datagrams successfully reassembled. Note that this counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the fragments.
OID	{ ipv6IfStatsEntry 17 }
NOTE	

OBJECT	ipv6IfStatsReasmFails
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of failures detected by the IPv6 re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IPv6 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. This counter is incremented at the interface to which these fragments were addressed which might not be

	necessarily the input interface for some of the fragments.
OID	{ ipv6IfStatsEntry 18 }
NOTE	

OBJECT	ipv6IfStatsInMcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of multicast packets received by the interface
OID	{ ipv6IfStatsEntry 19 }
NOTE	

OBJECT	ipv6IfStatsOutMcastPkts
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of multicast packets transmitted by the interface
OID	{ ipv6IfStatsEntry 20 }
NOTE	

OBJECT	ipv6AddrPrefixTable
SYNTAX	SEQUENCE OF Ipv6AddrPrefixEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The list of IPv6 address prefixes of IPv6 interfaces.
OID	{ ipv6MIBObjects 7 }
NOTE	

OBJECT	ipv6AddrPrefixEntry
SYNTAX	Ipv6AddrPrefixEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An interface entry containing objects of a particular IPv6 address prefix.
INDEX	ipv6IfIndex, ipv6AddrPrefix, ipv6AddrPrefixLength
OID	{ ipv6AddrPrefixTable 1 }
NOTE	

OBJECT	ipv6AddrPrefix
SYNTAX	Ipv6AddressPrefix
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The prefix associated with the this interface.
OID	{ ipv6AddrPrefixEntry 1 }
NOTE	

OBJECT	ipv6AddrPrefixLength
SYNTAX	INTEGER (0..128)
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The length of the prefix (in bits).
OID	{ ipv6AddrPrefixEntry 2 }
NOTE	

OBJECT	ipv6AddrPrefixOnLinkFlag
SYNTAX	TruthValue
MAX-ACCESS	read-only
STATUS	current
DSCRPT	This object has the value 'true(1)', if this prefix can be used for on-link determination and the value 'false(2)' otherwise.
OID	{ ipv6AddrPrefixEntry 3 }
NOTE	

OBJECT	ipv6AddrPrefixAutonomousFlag
SYNTAX	TruthValue
MAX-ACCESS	read-only
STATUS	current
DSCRPT	Autonomous address configuration flag. When true(1), indicates that this prefix can be used for autonomous address configuration (i.e. can be used to form a local interface address). If false(2), it is not used to autoconfigure a local interface address.
OID	{ ipv6AddrPrefixEntry 4 }
NOTE	

OBJECT	ipv6AddrPrefixAdvPreferredLifetime
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current

3-138

DSCRPT	It is the length of time in seconds that this prefix will remain preferred, i.e. time until deprecation. A value of 4,294,967,295 represents infinity. The address generated from a deprecated prefix should no longer be used as a source address in new communications, but packets received on such an interface are processed as expected.
OID	{ ipv6AddrPrefixEntry 5 }
NOTE	

OBJECT	ipv6AddrPrefixAdvValidLifetime
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	It is the length of time in seconds that this prefix will remain valid, i.e. time until invalidation. A value of 4,294,967,295 represents infinity. The address generated from an invalidated prefix should not appear as the destination or source address of a packet.
OID	{ ipv6AddrPrefixEntry 6 }
NOTE	

OBJECT	ipv6AddrTable
SYNTAX	SEQUENCE OF Ipv6AddrEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The table of addressing information relevant to this node's interface addresses.
OID	{ ipv6MIBObjects 8 }
NOTE	

OBJECT	ipv6AddrEntry
SYNTAX	Ipv6AddrEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The addressing information for one of this node's interface addresses.
INDEX	ipv6IfIndex, ipv6AddrAddress
OID	{ ipv6AddrTable 1 }

NOTE	

OBJECT	ipv6AddrAddress
SYNTAX	Ipv6Address
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The IPv6 address to which this entry's addressing information pertains.
OID	{ ipv6AddrEntry 1 }
NOTE	

OBJECT	ipv6AddrPfxLength
SYNTAX	INTEGER(0..128)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The length of the prefix (in bits) associated with the IPv6 address of this entry.
OID	{ ipv6AddrEntry 2 }
NOTE	

OBJECT	ipv6AddrType
SYNTAX	INTEGER { stateless(1), stateful(2), unknown(3) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The type of address. Note that 'stateless(1)' refers to an address that was statelessly autoconfigured; 'stateful(2)' refers to a address which was acquired by via a stateful protocol (e.g. DHCPv6, manual configuration).
OID	{ ipv6AddrEntry 3 }
NOTE	

OBJECT	ipv6AddrAnycastFlag
SYNTAX	TruthValue
MAX-ACCESS	read-only
STATUS	current
DSCRPT	This object has the value 'true(1)', if this address is an anycast address and the value 'false(2)' otherwise.
OID	{ ipv6AddrEntry 4 }
NOTE	

BJECT	ipv6AddrStatus

3-140

SYNTAX	INTEGER { preferred(1), deprecated(2), invalid(3), inaccessible(4), unknown(5) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	Address status. The preferred(1) state indicates that this is a valid address that can appear as the destination or source address of a packet. The deprecated(2) state indicates that this is a valid but deprecated address that should no longer be used as a source address in new communications, but packets addressed to such an address are processed as expected. The invalid(3) state indicates that this is not valid address which should not appear as the destination or source address of a packet. The inaccessible(4) state indicates that the address is not accessible because the interface to which this address is assigned is not operational.
OID	{ ipv6AddrEntry 5 }
NOTE	

OBJECT	ipv6RouteNumber
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of current ipv6RouteTable entries. This is primarily to avoid having to read the table in order to determine this number.
OID	{ ipv6MIBObjects 9 }
NOTE	

OBJECT	ipv6DiscardedRoutes
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of routing entries which 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.
OID	{ ipv6MIBObjects 10 }
NOTE	

OBJECT	ipv6RouteTable
SYNTAX	SEQUENCE OF Ipv6RouteEntry
MAX-ACCESS	not-accessible

STATUS	current
DSCRPT	IPv6 Routing table. This table contains an entry for each valid IPv6 unicast route that can be used for packet forwarding determination.
OID	{ ipv6MIBObjects 11 }
NOTE	

OBJECT	ipv6RouteEntry
SYNTAX	Ipv6RouteEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	A routing entry.
INDEX	ipv6RouteDest, ipv6RoutePfxLength, ipv6RouteIndex
OID	{ ipv6RouteTable 1 }
NOTE	

OBJECT	ipv6RouteDest
SYNTAX	Ipv6Address
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The destination IPv6 address of this route. This object may not take a Multicast address value.
OID	{ ipv6RouteEntry 1 }
NOTE	

OBJECT	ipv6RoutePfxLength
SYNTAX	INTEGER(0..128)
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	Indicates the prefix length of the destination address.
OID	{ ipv6RouteEntry 2 }
NOTE	

OBJECT	ipv6RouteIndex
SYNTAX	Unsigned32
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The value which uniquely identifies the route among the routes to the same network layer destination. The way this value is chosen is

	implementation specific but it must be unique for ipv6RouteDest/ipv6RoutePfxLength pair and remain constant for the life of the route.
OID	{ ipv6RouteEntry 3 }
NOTE	

OBJECT	ipv6RouteIfIndex
SYNTAX	Ipv6IfIndexOrZero
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The index value which 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 ipv6IfIndex. For routes of the discard type this value can be zero.
OID	{ ipv6RouteEntry 4 }
NOTE	

OBJECT	ipv6RouteNextHop
SYNTAX	Ipv6Address
MAX-ACCESS	read-only
STATUS	current
DSCRPT	On remote routes, the address of the next system en route; otherwise, ::0 ('00000000000000000000000000000000'H in ASN.1 string representation).
OID	{ ipv6RouteEntry 5 }
NOTE	

OBJECT	ipv6RouteType
SYNTAX	INTEGER { other(1), discard(2), local(3), remote(4) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The type of route. Note that 'local(3)' refers to a route for which the next hop is the final destination; 'remote(4)' refers to a route for which the next hop is not the final destination; 'discard(2)' refers to a route indicating that packets to destinations matching this route are to be discarded (sometimes called black-hole route).
OID	{ ipv6RouteEntry 6 }

NOTE	

OBJECT	ipv6RouteProtocol
SYNTAX	INTEGER { other(1), local(2), netmgmt(3), ndisc(4), rip(5), ospf(6), bgp(7), idrp(8), igrp(9) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The routing mechanism via which this route was learned.
OID	{ ipv6RouteEntry 7 }
NOTE	

OBJECT	ipv6RoutePolicy
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The general set of conditions that would cause the selection of one multipath route (set of next hops for a given destination) is referred to as 'policy'. Unless the mechanism indicated by ipv6RouteProtocol specified otherwise, the policy specifier is the 8-bit Traffic Class field of the IPv6 packet header that is zero extended at the left to a 32-bit value.

Protocols defining 'policy' otherwise must either define a set of values which are valid for this object or must implement an integer-instanced policy table for which this object's value acts as an index. |
| OID | { ipv6RouteEntry 8 } |
| NOTE | |

OBJECT	ipv6RouteAge
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of `too old' can be implied except through knowledge of the routing protocol by which the route was learned.
OID	{ ipv6RouteEntry 9 }
NOTE	

OBJECT	ipv6RouteNextHopRDI

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The Routing Domain ID of the Next Hop. The semantics of this object are determined by the routing-protocol specified in the route's ipv6RouteProtocol value. When this object is unknown or not relevant its value should be set to zero.
OID	{ ipv6RouteEntry 10 }
NOTE	

OBJECT	ipv6RouteMetric
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipv6RouteProtocol value. When this is unknown or not relevant to the protocol indicated by ipv6RouteProtocol, the object value should be set to its maximum value (4,294,967,295).
OID	{ ipv6RouteEntry 11 }
NOTE	

OBJECT	ipv6RouteWeight
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The system internal weight value for this route. The semantics of this value are determined by the implementation specific rules. Generally, within routes with the same ipv6RoutePolicy value, the lower the weight value the more preferred is the route.
OID	{ ipv6RouteEntry 12 }
NOTE	

OBJECT	ipv6RouteInfo
SYNTAX	RowPointer
MAX-ACCESS	read-only
STATUS	current
DSCRPT	A reference to MIB definitions specific to the

	particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipv6RouteProto value. If this information is not present, its value should be set to the OBJECT ID { 0 0 }, which is a syntactically valid object identifier, and any implementation conforming to ASN.1 and the Basic Encoding Rules must be able to generate and recognize this value.
OID	{ ipv6RouteEntry 13 }
NOTE	

OBJECT	ipv6RouteValid
SYNTAX	TruthValue
MAX-ACCESS	read-write
STATUS	current
DSCRPT	Setting this object to the value 'false(2)' has the effect of invalidating the corresponding entry in the ipv6RouteTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said 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 ipv6RouteValid object.
OID	{ ipv6RouteEntry 14 }
NOTE	

OBJECT	ipv6NetToMediaTable
SYNTAX	SEQUENCE OF Ipv6NetToMediaEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The IPv6 Address Translation table used for mapping from IPv6 addresses to physical addresses. The IPv6 address translation table contain the Ipv6Address to `physical' address equivalencies. Some interfaces do not use translation tables for determining address equivalencies; if all interfaces are of this type, then the Address Translation table is empty, i.e., has zero entries.

3-146

OID	{ ipv6MIBObjects 12 }
NOTE	

OBJECT	ipv6NetToMediaEntry
SYNTAX	Ipv6NetToMediaEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	Each entry contains one IPv6 address to `physical' address equivalence.
INDEX	ipv6IfIndex, ipv6NetToMediaNetAddress
OID	{ ipv6NetToMediaTable 1 }
NOTE	

OBJECT	ipv6NetToMediaNetAddress
SYNTAX	Ipv6Address
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The IPv6 Address corresponding to the media-dependent `physical' address.
OID	{ ipv6NetToMediaEntry 1 }
NOTE	

OBJECT	ipv6NetToMediaPhysAddress
SYNTAX	PhysAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The media-dependent `physical' address.
OID	{ ipv6NetToMediaEntry 2 }
NOTE	

OBJECT	ipv6NetToMediaType
SYNTAX	INTEGER { other(1), dynamic(2), static(3), local(4) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The type of the mapping. The 'dynamic(2)' type indicates that the IPv6 address to physical addresses mapping has been dynamically resolved using the IPv6 Neighbor Discovery protocol. The static(3)' types indicates that the mapping has been statically configured. The local(4) indicates that the mapping is provided for an entity's own interface address.
OID	{ ipv6NetToMediaEntry 3 }

NOTE	

OBJECT	ipv6IfNetToMediaState
SYNTAX	INTEGER { reachable(1), stale(2), delay(3), probe(4), invalid(5), unknown(6) }
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The Neighbor Unreachability Detection [8] state for the interface when the address mapping in this entry is used.
OID	{ ipv6NetToMediaEntry 4 }
NOTE	

OBJECT	ipv6IfNetToMediaLastUpdated
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The value of sysUpTime at the time this entry was last updated. If this entry was updated prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.
OID	{ ipv6NetToMediaEntry 5 }
NOTE	

OBJECT	ipv6NetToMediaValid
SYNTAX	TruthValue
MAX-ACCESS	read-write
STATUS	current
DSCRPT	Setting this object to the value 'false(2)' has the effect of invalidating the corresponding entry in the ipv6NetToMediaTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said 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 ipv6NetToMediaValid object.
OID	{ ipv6NetToMediaEntry 6 }
NOTE	

TRAP	ipv6IfStateChange
OBJECTS	ipv6IfDescr, ipv6IfOperStatus -- the new state of the If.
STATUS	current
DSCRPT	An ipv6IfStateChange notification signifies that there has been a change in the state of an ipv6 interface. This notification should be generated when the interface's operational status transitions to or from the up(1) state.
OID	{ ipv6NotificationPrefix 1 }
NOTE	

COMPLIANCE	ipv6Compliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement ipv6 MIB.
MANDATORY-G	ipv6GeneralGroup, ipv6NotificationGroup
OBJECT	ipv6Forwarding
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6DefaultHopLimit
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6IfDescr
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6IfIdentifier
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6IfIdentifierLength
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6IfAdminStatus
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OBJECT	ipv6RouteValid
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object

OBJECT	ipv6NetToMediaValid
MIN-ACCESS	read-only
DSCRPT	An agent is not required to provide write access to this object
OID	{ ipv6Compliances 1 }
NOTE	

OBJECT-G	ipv6GeneralGroup
OBJECTS	ipv6Forwarding, ipv6DefaultHopLimit, ipv6Interfaces, ipv6IfTableLastChange, ipv6IfDescr, ipv6IfLowerLayer, ipv6IfEffectiveMtu, ipv6IfReasmMaxSize, ipv6IfIdentifier, ipv6IfIdentifierLength, ipv6IfPhysicalAddress, ipv6IfAdminStatus, ipv6IfOperStatus, ipv6IfLastChange, ipv6IfStatsInReceives, ipv6IfStatsInHdrErrors, ipv6IfStatsInTooBigErrors, ipv6IfStatsInNoRoutes, ipv6IfStatsInAddrErrors, ipv6IfStatsInUnknownProtos, ipv6IfStatsInTruncatedPkts, ipv6IfStatsInDiscards, ipv6IfStatsInDelivers, ipv6IfStatsOutForwDatagrams, ipv6IfStatsOutRequests, ipv6IfStatsOutDiscards, ipv6IfStatsOutFragOKs, ipv6IfStatsOutFragFails, ipv6IfStatsOutFragCreates, ipv6IfStatsReasmReqds, ipv6IfStatsReasmOKs, ipv6IfStatsReasmFails, ipv6IfStatsInMcastPkts, ipv6IfStatsOutMcastPkts, ipv6AddrPrefixOnLinkFlag, ipv6AddrPrefixAutonomousFlag, ipv6AddrPrefixAdvPreferredLifetime, ipv6AddrPrefixAdvValidLifetime, ipv6AddrPfxLength, ipv6AddrType, ipv6AddrAnycastFlag, ipv6AddrStatus, ipv6RouteNumber, ipv6DiscardedRoutes, ipv6RouteIfIndex, ipv6RouteNextHop, ipv6RouteType, ipv6RouteProtocol, ipv6RoutePolicy, ipv6RouteAge, ipv6RouteNextHopRDI, ipv6RouteMetric, ipv6RouteWeight, ipv6RouteInfo, ipv6RouteValid, ipv6NetToMediaPhysAddress, ipv6NetToMediaType, ipv6IfNetToMediaState, ipv6IfNetToMediaLastUpdated, ipv6NetToMediaValid
STATUS	current
DSCRPT	The IPv6 group of objects providing for basic management of IPv6 entities.
OID	{ ipv6Groups 1 }
NOTE	

NOTIFICATION-G	ipv6NotificationGroup
NOTIFICATIONS	ipv6IfStateChange
STATUS	current
DSCRPT	The notification that an IPv6 entity is required to implement.
OID	{ ipv6Groups 2 }
NOTE	

3.13 IPV6-ICMP-MIB

MODULE	ipv6IcmpMIB
UPDATED	9801082155Z
DSCRPT	The MIB module for entities implementing the ICMPv6.
OID	{ mib-2 56 }
NOTE	

OBJECT	ipv6IfIcmpTable
SYNTAX	SEQUENCE OF Ipv6IfIcmpEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	IPv6 ICMP statistics. This table contains statistics of ICMPv6 messages that are received and sourced by the entity.
OID	{ ipv6IcmpMIBObjects 1 }
NOTE	

OBJECT	ipv6IfIcmpEntry
SYNTAX	Ipv6IfIcmpEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	An ICMPv6 statistics entry containing objects at a particular IPv6 interface. Note that a receiving interface is the interface to which a given ICMPv6 message is addressed which may not be necessarily the input interface for the message. Similarly, the sending interface is the interface that sources a given ICMP message which is usually but not necessarily the output interface for the message.
AUGMENTS	ipv6IfEntry
OID	{ ipv6IfIcmpTable 1 }
NOTE	

OBJECT	ipv6IfIcmpInMsgs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of ICMP messages received

	by the interface which includes all those counted by ipv6IfIcmpInErrors. Note that this interface is the interface to which the ICMP messages were addressed which may not be necessarily the input interface for the messages.
OID	{ ipv6IfIcmpEntry 1 }
NOTE	

OBJECT	ipv6IfIcmpInErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP messages which the interface received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).
OID	{ ipv6IfIcmpEntry 2 }
NOTE	

OBJECT	ipv6IfIcmpInDestUnreachs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Destination Unreachable messages received by the interface.
OID	{ ipv6IfIcmpEntry 3 }
NOTE	

OBJECT	ipv6IfIcmpInAdminProhibs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP destination unreachable/communication administratively prohibited messages received by the interface.
OID	{ ipv6IfIcmpEntry 4 }
NOTE	

OBJECT	ipv6IfIcmpInTimeExcds
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Time Exceeded messages received by the interface.

OID	{ ipv6IfIcmpEntry 5 }
NOTE	

OBJECT	ipv6IfIcmpInParmProblems
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Parameter Problem messages received by the interface.
OID	{ ipv6IfIcmpEntry 6 }
NOTE	

OBJECT	ipv6IfIcmpInPktTooBigs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Packet Too Big messages received by the interface.
OID	{ ipv6IfIcmpEntry 7 }
NOTE	

OBJECT	ipv6IfIcmpInEchos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo (request) messages received by the interface.
OID	{ ipv6IfIcmpEntry 8 }
NOTE	

OBJECT	ipv6IfIcmpInEchoReplies
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo Reply messages received by the interface.
OID	{ ipv6IfIcmpEntry 9 }
NOTE	

OBJECT	ipv6IfIcmpInRouterSolicits
SYNTAX	Counter32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	The number of ICMP Router Solicit messages received by the interface.
OID	{ ipv6IfIcmpEntry 10 }
NOTE	

OBJECT	ipv6IfIcmpInRouterAdvertisements
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Router Advertisement messages received by the interface.
OID	{ ipv6IfIcmpEntry 11 }
NOTE	

OBJECT	ipv6IfIcmpInNeighborSolicits
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Neighbor Solicit messages received by the interface.
OID	{ ipv6IfIcmpEntry 12 }
NOTE	

OBJECT	ipv6IfIcmpInNeighborAdvertisements
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Neighbor Advertisement messages received by the interface.
OID	{ ipv6IfIcmpEntry 13 }
NOTE	

OBJECT	ipv6IfIcmpInRedirects
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of Redirect messages received by the interface.
OID	{ ipv6IfIcmpEntry 14 }
NOTE	

OBJECT	ipv6IfIcmpInGroupMembQueries
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Query messages received by the interface.
CID	{ ipv6IfIcmpEntry 15}
NOTE	

OBJECT	ipv6IfIcmpInGroupMembResponses
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Response messages received by the interface.
CID	{ ipv6IfIcmpEntry 16}
NOTE	

OBJECT	ipv6IfIcmpInGroupMembReductions
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Reduction messages received by the interface.
CID	{ ipv6IfIcmpEntry 17}
NOTE	

OBJECT	ipv6IfIcmpOutMsgs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The total number of ICMP messages which this interface attempted to send. Note that this counter includes all those counted by icmpOutErrors.
CID	{ ipv6IfIcmpEntry 18 }
NOTE	

OBJECT	ipv6IfIcmpOutErrors
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP messages which this interface 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 IPv6 to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.
CID	{ ipv6IfIcmpEntry 19 }
NOTE	

OBJECT	ipv6IfIcmpOutDestUnreachs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Destination Unreachable messages sent by the interface.
CID	{ ipv6IfIcmpEntry 20 }
NOTE	

OBJECT	ipv6IfIcmpOutAdminProhibs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	Number of ICMP dest unreachable/communication administratively prohibited messages sent.
CID	{ ipv6IfIcmpEntry 21 }
NOTE	

OBJECT	ipv6IfIcmpOutTimeExcds
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Time Exceeded messages sent by the interface.
CID	{ ipv6IfIcmpEntry 22 }
NOTE	

OBJECT	ipv6IfIcmpOutParmProblems
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Parameter Problem messages sent by the interface.
CID	{ ipv6IfIcmpEntry 23 }

NOTE	

OBJECT	ipv6IfIcmpOutPktTooBigs
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Packet Too Big messages sent by the interface.
CID	{ ipv6IfIcmpEntry 24 }
NOTE	

OBJECT	ipv6IfIcmpOutEchos
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo (request) messages sent by the interface.
CID	{ ipv6IfIcmpEntry 25 }
NOTE	

OBJECT	ipv6IfIcmpOutEchoReplies
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Echo Reply messages sent by the interface.
CID	{ ipv6IfIcmpEntry 26 }
NOTE	

OBJECT	ipv6IfIcmpOutRouterSolicits
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Router Solicitation messages sent by the interface.
CID	{ ipv6IfIcmpEntry 27 }
NOTE	

OBJECT	ipv6IfIcmpOutRouterAdvertisements
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current

DSCRPT	The number of ICMP Router Advertisement messages sent by the interface.
OID	{ ipv6IfIcmpEntry 28 }
NOTE	

OBJECT	ipv6IfIcmpOutNeighborSolicits
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Neighbor Solicitation messages sent by the interface.
OID	{ ipv6IfIcmpEntry 29 }
NOTE	

OBJECT	ipv6IfIcmpOutNeighborAdvertisements
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMP Neighbor Advertisement messages sent by the interface.
OID	{ ipv6IfIcmpEntry 30 }
NOTE	

OBJECT	ipv6IfIcmpOutRedirects
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.
OID	{ ipv6IfIcmpEntry 31 }
NOTE	

OBJECT	ipv6IfIcmpOutGroupMembQueries
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Query messages sent.
OID	{ ipv6IfIcmpEntry 32}
NOTE	

OBJECT	ipv6IfIcmpOutGroupMembResponses

3-158

SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Response messages sent.
CID	{ ipv6IfIcmpEntry 33}
NOTE	

OBJECT	ipv6IfIcmpOutGroupMembReductions
SYNTAX	Counter32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	The number of ICMPv6 Group Membership Reduction messages sent.
CID	{ ipv6IfIcmpEntry 34}
NOTE	

COMPLIANCE	ipv6IcmpCompliance
STATUS	current
DSCRPT	The compliance statement for SNMPv2 entities which implement ICMPv6.
MANDATORY-G	ipv6IcmpGroup
CID	{ ipv6IcmpCompliances 1 }
NOTE	

OBJECT-G	ipv6IcmpGroup
OBJECTS	ipv6IfIcmpInMsgs, ipv6IfIcmpInErrors, ipv6IfIcmpInDestUnreachs, ipv6IfIcmpInAdminProhibs, ipv6IfIcmpInTimeExcds, ipv6IfIcmpInParmProblems, ipv6IfIcmpInPktTooBigs, ipv6IfIcmpInEchos, ipv6IfIcmpInEchoReplies, ipv6IfIcmpInRouterSolicits, ipv6IfIcmpInRouterAdvertisements, ipv6IfIcmpInNeighborSolicits, ipv6IfIcmpInNeighborAdvertisements, ipv6IfIcmpInRedirects, ipv6IfIcmpInGroupMembQueries, ipv6IfIcmpInGroupMembResponses, ipv6IfIcmpInGroupMembReductions, ipv6IfIcmpOutMsgs, ipv6IfIcmpOutErrors, ipv6IfIcmpOutDestUnreachs, ipv6IfIcmpOutAdminProhibs, ipv6IfIcmpOutTimeExcds, ipv6IfIcmpOutParmProblems, ipv6IfIcmpOutPktTooBigs, ipv6IfIcmpOutEchos, ipv6IfIcmpOutEchoReplies, ipv6IfIcmpOutRouterSolicits, ipv6IfIcmpOutRouterAdvertisements, ipv6IfIcmpOutNeighborSolicits, ipv6IfIcmpOutNeighborAdvertisements, ipv6IfIcmpOutRedirects, ipv6IfIcmpOutGroupMembQueries, ipv6IfIcmpOutGroupMembResponses, ipv6IfIcmpOutGroupMembReductions
STATUS	current
DSCRPT	The ICMPv6 group of objects providing information specific to ICMPv6.

OID	{ ipv6IcmpGroups 1 }
NOTE	

3.14 IANAifType-MIB

MODULE	ianaifType
UPDATED	200105110000Z
DSCRPT	This MIB module defines the IANAifType Textual Convention, and thus the enumerated values of the ifType object defined in MIB-II's ifTable.
REVISION	200105110000Z
DSCRPT	Registration of new IANAifType197.
REVISION	200101120000Z
DSCRPT	Registration of new IANAifTypes195 and 196.
REVISION	200012190000Z
DSCRPT	Registration of new IANAifTypes193 and 194.
REVISION	200012070000Z
DSCRPT	Registration of new IANAifTypes191 and 192.
REVISION	200012040000Z
DSCRPT	Registration of new IANAifType190.
REVISION	200010170000Z
DSCRPT	Registration of new IANAifTypes188 and 189.
REVISION	200010020000Z
DSCRPT	Registration of new IANAifType 187.
REVISION	200009010000Z
DSCRPT	Registration of new IANAifTypes184, 185, and 186.
REVISION	200008240000Z
DSCRPT	Registration of new IANAifType 183.
REVISION	200008230000Z
DSCRPT	Registration of new IANAifTypes174-182.
REVISION	200008220000Z
DSCRPT	Registration of new IANAifTypes 170, 171, 172 and 173.
REVISION	200004250000Z
DSCRPT	Registration of new IANAifTypes 168 and 169.
REVISION	200003060000Z
DSCRPT	Fixed a missing semi-colon in the IMPORT. Also cleaned up the REVISION log a bit. It is not complete, but from now on it will be maintained and kept up to date with each change to this MIB module.
REVISION	199910081430Z
DSCRPT	Include new name assignments up to cnr(85). This is the first version

	available via the WWW at: ftp://ftp.isi.edu/mib/ianaiftype.mib
REVISION	199401310000Z
DSCRPT	Initial version of this MIB as published in RFC 1573.
OID	{ mib-2 30 }
NOTE	

TC	IANAifType
STATUS	current
DSCRPT	This data type is used as the syntax of the ifType object in the (updated) definition of MIB-II's ifTable. The definition of this textual convention with the addition of newly assigned values is published periodically by the IANA, in either the Assigned Numbers RFC, or some derivative of it specific to Internet Network Management number assignments. (The latest arrangements can be obtained by contacting the IANA.) Requests for new values should be made to IANA via email (iana@iana.org). The relationship between the assignment of ifType values and of OIDs to particular media-specific MIBs is solely the purview of IANA and is subject to change without notice. Quite often, a media-specific MIB's OID-subtree assignment within MIB-II's 'transmission' subtree will be the same as its ifType value. However, in some circumstances this will not be the case, and implementors must not pre-assume any specific relationship between ifType values and transmission subtree OIDs.
SYNTAX	INTEGER { other(1), regular1822(2), hdh1822(3), ddnX25(4), rfc877x25(5), ethernetCsmacd(6), iso88023Csmacd(7), iso88024TokenBus(8), iso88025TokenRing(9), iso88026Man(10), starLan(11), proteon10Mbit(12), proteon80Mbit(13), hyperchannel(14), fddi(15), lapb(16), sdlc(17), ds1(18), e1(19), basicISDN(20), primaryISDN(21), propPointToPointSerial(22), ppp(23), softwareLoopback(24), eon(25), ethernet3Mbit(26), nsip(27), slip(28), ultra(29), ds3(30), sip(31), frameRelay(32), rs232(33), para(34), arcnet(35), arcnetPlus(36), atm(37), miox25(38), sonet(39), x25ple(40), iso88022llc(41), localTalk(42), smdsDxi(43), frameRelayService(44), v35(45), hssi(46), hippi(47), modem(48), aal5(49), sonetPath(50), sonetVT(51), smdsIcip(52), propVirtual(53), propMultiplexor(54), ieee80212(55), fibreChannel(56), hippiInterface(57), frameRelayInterconnect(58), aflane8023(59), aflane8025(60), cctEmul(61), fastEther(62), isdn(63),

	v11(64), v36(65), g703at64k(66), g703at2mb(67), qllc(68), fastEtherFX(69), channel(70), ieee80211(71), ibm370parChan(72), escon(73), dlsw(74), isdns(75), isdnu(76), lapd(77), ipSwitch(78), rsrb(79), atmLogical(80), ds0(81), ds0Bundle(82), bsc(83), async(84), cnr(85), iso88025Dtr(86), eplrs(87), arap(88), propCnls(89), hostPad(90), termPad(91), frameRelayMPI(92), x213(93), adsl(94), radsl(95), sdsl(96), vdsl(97), iso88025CRFPInt(98), myrinet(99), voiceEM(100), voiceFXO(101), voiceFXS(102), voiceEncap(103), voiceOverIp(104), atmDxi(105), atmFuni(106), atmIma (107), pppMultilinkBundle(108), ipOverCdlc (109), ipOverClaw (110), stackToStack (111), virtualIpAddress (112), mpc (113), ipOverAtm(114), iso88025Fiber (115), tdlc (116), gigabitEthernet (117), hdlc (118), lapf (119), v37 (120), x25mlp (121), x25huntGroup (122), trasnpHdlc (123), interleave (124), fast (125), ip (126), docsCableMaclayer (127), docsCableDownstream(128), docsCableUpstream(129), a12MppSwitch (130), tunnel (131), coffee (132), ces (133), atmSubInterface (134), l2vlan (135), l3ipvlan (136), l3ipxvlan (137), digitalPowerline (138), mediaMailOverIp (139), dtm(140), dcn (141), ipForward (142), msdsl (143), ieee1394 (144), if-gsn (145), dvbRccMacLayer (146), dvbRccDownstream (147), dvbRccUpstream(148), atmVirtual (149), mplsTunnel (150), srp (151), voiceOverAtm(152), voiceOverFrameRelay (153), idsl (154), compositeLink (155), ss7SigLink (156), propWrelessP2P (157), frForward (158), rfc1483 (159), usb (160), ieee8023adLag (161), bgppolicyaccounting (162), frf16MfrBundle (163), h323Gatekeeper (164), h323Proxy (165), mpls (166), mfSigLink (167), hdsl2 (168), shdsl (169), ds1FDL (170), pos (171), dvbAsiIn (172), dvbAsiOut (173), plc (174), nfas (175), tr008 (176), gr303RDT (177), gr303IDT (178), isup (179), propDocsWrelessMaclayer (180), propDocsWrelessDownstream(181), propDocsWrelessUpstream(182), hiperlan2 (183), propBWAp2Mp (184), sonetOverheadChannel (185), digitalWapperOverheadChannel (186), aal2 (187), radioMAC (188), atmRadio (189), imt (190), mvl (191), reachDSL (192), frDlciEndPt (193), atmVciEndPt (194), opticalChannel (195), opticalTransport (196), propAtm (197) }

3.15 INET-ADDRESS-MIB

MODULE	inetAddressMIB
UPDATED	200205090000Z
DSCRPT	This MIB module defines textual conventions for representing Internet addresses. An Internet address can be an IPv4 address, an IPv6 address or a DNS domain name. This module also defines textual conventions for Internet port numbers, autonomous system numbers and the length of an Internet address prefix.
REVISION	200205090000Z
DSCRPT	Second version, published as RFC 3291. This revisions contains several clarifications and it

	introduces several new textual conventions: InetAddressPrefixLength, InetPortNumber, InetAutonomousSystemNumber, InetAddressIPv4z, and InetAddressIPv6z.
REVISION	200006080000Z
DSCRPT	Initial version, published as RFC 2851.
OID	{ mib-2 76 }
NOTE	

TC	InetAddressType
STATUS	current
DSCRPT	A value that represents a type of Internet address. unknown(0) An unknown address type. This value MUST be used if the value of the corresponding InetAddress object is a zero-length string. It may also be used to indicate an IP address which is not in one of the formats defined below ipv4(1) An IPv4 address as defined by the InetAddressIPv4 textual convention. ipv6(2) A global IPv6 address as defined by the InetAddressIPv6 textual convention. ipv4z(3) A non-global IPv4 address including a zone index as defined by the InetAddressIPv4z textual convention. ipv6z(4) A non-global IPv6 address including a zone index as defined by the InetAddressIPv6z textual convention. dns(16) A DNS domain name as defined by the InetAddressDNS textual convention. Each definition of a concrete InetAddressType value must be accompanied by a definition of a textual convention for use with that InetAddressType. To support future extensions, the InetAddressType textual convention SHOULD NOT be sub-typed in object type definitions. It MAY be sub-typed in compliance statements in order to require only a subset of these address types for a compliant implementation. Implementations must ensure that InetAddressType objects and any dependent objects (e.g. InetAddress objects) are

	consistent. An inconsistentValue error must be generated if an attempt to change an InetAddressType object would, for example, lead to an undefined InetAddress value. In particular, InetAddressType/InetAddress pairs must be changed together if the address type changes (e.g. from ipv6(2) to ipv4(1)).
SYNTAX	INTEGER { unknown(0), ipv4(1), ipv6(2), ipv4z(3), ipv6z(4), dns(16) }

TC	InetAddress
STATUS	current
DSCRPT	Denotes a generic Internet address. An InetAddress value is always interpreted within the context of an InetAddressType value. Every usage of the InetAddress textual convention is required to specify the InetAddressType object which provides the context. It is suggested that the InetAddressType object is logically registered before the object(s) which use the InetAddress textual convention if they appear in the same logical row The value of an InetAddress object must always be consistent with the value of the associated InetAddressType object. Attempts to set an InetAddress object to a value which is inconsistent with the associated InetAddressType must fail with an inconsistentValue error. When this textual convention is used as the syntax of an index object, there may be issues with the limit of 128 sub-identifiers specified in SMIv2, STD 58. In this case, the object definition MUST include a 'SIZE' clause to limit the number of potential instance sub-identifiers.
SYNTAX	OCTET STRING (SIZE (0..255))

TC	InetAddressIPv4
STATUS	current
DSCRPT	Represents an IPv4 network address: octets contents encoding 1-4 IPv4 address network-byte order The corresponding InetAddressType value is ipv4(1). This textual convention SHOULD NOT be used directly in object definitions since it restricts addresses to a specific format. However, if it is used, it MAY be used either on its own or in conjunction with InetAddressType as a pair.
SYNTAX	OCTET STRING (SIZE (4))

TC	InetAddressIPv6
STATUS	current
DSCRPT	Represents an IPv6 network address: octets contents encoding 1-16 IPv6 address network-byte order The corresponding InetAddressType value is ipv6(2). This textual convention SHOULD NOT be used directly in object definitions since it restricts addresses to a specific format. However, if it is used, it MAY be used either on its own or in conjunction with InetAddressType as a pair.
SYNTAX	OCTET STRING (SIZE (16))

TC	InetAddressIPv4z
STATUS	current
DSCRPT	Represents a non-global IPv4 network address together with its zone index: octets contents encoding 1-4 IPv4 address network-byte order 5-8 zone index network-byte order The corresponding InetAddressType value is ipv4z(3). The zone index (bytes 5-8) is used to disambiguate identical address values on nodes which have interfaces attached to different zones of the same scope. The zone index may contain the special value 0 which refers to the default zone for each scope. This textual convention SHOULD NOT be used directly in object definitions since it restricts addresses to a specific format. However, if it is used, it MAY be used either on its own or in conjunction with InetAddressType as a pair.
SYNTAX	OCTET STRING (SIZE (8))

TC	InetAddressIPv6z
STATUS	current
DSCRPT	Represents a non-global IPv6 network address together with its zone index: octets contents encoding 1-16 IPv6 address network-byte order 17-20 zone index network-byte order The corresponding InetAddressType value is ipv6z(4).

The zone index (bytes 17-20) is used to disambiguate
identical address values on nodes which have interfaces
attached to different zones of the same scope. The zone index
may contain the special value 0 which refers to the default
zone for each scope.

This textual convention SHOULD NOT be used directly in object
definitions since it restricts addresses to a specific format.
However, if it is used, it MAY be used either on its own or in
conjunction with InetAddressType as a pair.

SYNTAX	OCTET STRING (SIZE (20))

TC	InetAddressDNS
STATUS	current
DSCRPT	Represents a DNS domain name. The name SHOULD be fully qualified whenever possible.
	The corresponding InetAddressType is dns(16).
	The DESCRIPTION clause of InetAddress objects that may have InetAddressDNS values must fully describe how (and when) such names are to be resolved to IP addresses.
	This textual convention SHOULD NOT be used directly in object definitions since it restricts addresses to a specific format. However, if it is used, it MAY be used either on its own or in conjunction with InetAddressType as a pair.
SYNTAX	OCTET STRING (SIZE (1..255))

TC	InetAddressPrefixLength
STATUS	current
DSCRPT	Denotes the length of a generic Internet network address prefix. A value of n corresponds to an IP address mask which has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0.
	An InetAddressPrefixLength value is always interpreted within the context of an InetAddressType value. Every usage of the InetAddressPrefixLength textual convention is required to specify the InetAddressType object which provides the context. It is suggested that the InetAddressType object is logically registered before the object(s) which use the InetAddressPrefixLength textual convention if they appear in the same logical row
	InetAddressPrefixLength values that are larger than the maximum length of an IP address for a specific

	InetAddressType are treated as the maximum significant value applicable for the InetAddressType. The maximum significant value is 32 for the InetAddressType 'ipv4(1)' and 'ipv4z(3)' and 128 for the InetAddressType 'ipv6(2)' and 'ipv6z(4)'. The maximum significant value for the InetAddressType 'dns(16)' is 0. The value zero is object-specific and must be defined as part of the description of any object which uses this syntax. Examples of the usage of zero might include situations where the Internet network address prefix is unknown or does not apply.
SYNTAX	Unsigned32

TC	InetPortNumber
STATUS	current
DSCRPT	Represents a 16 bit port number of an Internet transport layer protocol. Port numbers are assigned by IANA. A current list of all assignments is available from . The value zero is object-specific and must be defined as part of the description of any object which uses this syntax. Examples of the usage of zero might include situations where a port number is unknown, or when the value zero is used as a wildcard in a filter.
SYNTAX	Unsigned32 (0..65535)

TC	InetAutonomousSystemNumber
STATUS	current
DSCRPT	Represents an autonomous system number which identifies an Autonomous System (AS). An AS is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs'. IANA maintains the AS number space and has delegated large parts to the regional registries. Autonomous system numbers are currently limited to 16 bits (0..65535). There is however work in progress to enlarge the autonomous system number space to 32 bits. This textual convention therefore uses an Unsigned32 value without a range restriction in order to support a larger autonomous system number space.
SYNTAX	Unsigned32

3.16 FURUKAWA-INFONETBASE-MIB

OBJECT	infResetSystem
SYNTAX	INTEGER { none(1), reset(2), indicated-reset(3), unindicated-reset(4) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	, 1)　　　　　　　MIB　GET　　　　　　, 1 2)　　　　　　　　　　　　　　　　　　　　　　　　　　　　, 　 MIB 2　SET　　　　　　　, 　 MIB　 2　　　, 　　　　　,　　　　　　　　　　　　　(infReservedIpAddress　　)　　　　　　　(　　) 　　　　　　Coldstart　　　　　　　　　, 　　 MIB　 3 Coldstart　　　　　　　　　　　　　　 (　　 1) 3)　　　　　　　　　　　　　　　　　　, 　　　　　　　　　　　　　, 　　　 ,　　　　　　　　　　　　　(infReservedIpAddress　　)　　　　　　(　　) 　　　　　　Coldstart　　　　　　　　　, 　　 MIB　 3 Coldstart　　　　　　　　　　　　　　 (　　 1) 4)　　　　　　　　　　　　　　　　　, 　　　　　 , 　　　 ,　　　　　　　　　　　　　(infReservedIpAddress　　)　　　　　　(　　　　　) 　　　　　Coldstart　　　　　　　　　, 　　 MIB　 4 Coldstart　　　　　　　　　　　　　 (　　 1) 5)　　　　　　　 (　　　　　　　　　　　　 : 　　　　) 　　　　　　 , 　　　 ,　　　　　　　　　　　　　(infReservedIpAddress　　)　　　　　 (　　　　　) 　　　　　Coldstart　　　　　　　　, 　　 MIB Coldstart　　　　　　　　　　　　 (　　 1)
OID	{ infonetSystem 1 }
NOTE	

OBJECT	infSaveConfig
SYNTAX	INTEGER { save-equal(1), different(2) }

ACCESS	read-write
STATUS	optional
DSCRPT	SET 1 1 SET , , ( ) GET , 1) 1 , ( ) , 2) 2 , ( ) ,
OID	{ infonetSystem 2 }
NOTE	

OBJECT	infRestoreConfig
SYNTAX	INTEGER { restore-equal(1), different(2) }
ACCESS	read-write
STATUS	optional
DSCRPT	SET 1 1 SET , (,) GET , 1) 1 , ( ) , 2) 2 , ( ) ,
OID	{ infonetSystem 3 }
NOTE	

OBJECT	infSlotCapacity
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	( )
OID	{ infonetSystem 4 }
NOTE	

OBJECT	infSlotMap
SYNTAX	OCTET STRING (SIZE (4))
ACCESS	read-write
STATUS	mandatory
DSCRPT	, (1) , (infonet) ,

	0 : ( ),
	1 :
	MIB X'B0000000' , 1, 3, 4
	2 5 ,
OID	{ infonetSystem 5 }
NOTE	

OBJECT	infCurrentTime
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-write
STATUS	optional

DSCRPT	. . . . : :
	
	                    MIB
	
	                                   ,
	
	    4
	 , , , , 2
	        3
	(sun mon tue wed thu fri sat)
	
	1991.12.13.fri.19:21:45
	1992.01.05.sun.03:09:07

OID	{ infonetSystem 6 }
NOTE	

OBJECT	infReservedIpAddress
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	SNMP IP , (MIB-II)
OID	{ infonetSystem 7 }
NOTE	

OBJECT	infReservedSubnetMask
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	SNMP , (MIB-II

	)
OID	{ infonetSystem 8 }
NOTE	

OBJECT	infSystemSelfTest
SYNTAX	INTEGER { none(1), do-selftest(2) }
ACCESS	read-write
STATUS	optional
DSCRPT	, 2 SE T 2 SET , MIB 2 , 1 , GET , MIB 1
OID	{ infonetSystem 9 }
NOTE	

OBJECT	infSystemSelfTestResult
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	, , ,
OID	{ infonetSystem 10 }
NOTE	

OBJECT	infMgrTable
SYNTAX	SEQUENCE OF InfMgrEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	
OID	{ infonetSystem 11 }
NOTE	

OBJECT	infMgrEntry
SYNTAX	InfMgrEntry

ACCESS	not-accessible
STATUS	mandatory
DSCRPT	！
INDEX	infMgrIndex
OID	{ infMgrTable 1 }
NOTE	

OBJECT	infMgrIndex
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
DSCRPT	, IP IP
OID	{ infMgrEntry 1 }
NOTE	

OBJECT	infMgrIpAddress
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	IP
OID	{ infMgrEntry 2 }
NOTE	

OBJECT	infMgrCommunityName
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-write
STATUS	mandatory
DSCRPT	
OID	{ infMgrEntry 3 }
NOTE	

OBJECT	infMgrType
SYNTAX	INTEGER { other(1), reserved(2), acceptset-sendtrap(3), acceptset-notsendtrap(4), notacceptset-sendtrap(5), notacceptset-notsendtrap(6) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	SET( ) , 1： , 2： ,

	3 :　　　　　　　　　　　　　, 　　　　　　　　　　　　, SET 　　　　　　　　　　　　, GET 4 :　　　　　　　　　　　　　, 　　　　　　　　　　　　, SET 　　　　　　　　　　　　, GET 5 :　　　　　　　　　　　　　, 　　　　　　　　　　　　, SET 　　　　　　　　, infMgrTable　　　　　　　　　　, infMgrTable　3　　　4　　　　　　　　　　　GET 　　　　MIB　　　　　　, GET 6 :　　　　　　　　　　　　　, 　　　　　　　　　　　　, SET 　　　　　　　　, infMgrTable　　　　　　　　　　, infMgrTable　3　　　4　　　　　　　　　　　GET 　　　　MIB　　　　　　, GET
OID	{ infMgrEntry 4 }
NOTE	

OBJECT	infMgrStatus
SYNTAX	INTEGER { other(1), invalid(2), permanent(3) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	 1 :　　　 2, 3 2 :　　　　　 (　　　　　　) 3 :
OID	{ infMgrEntry 5 }
NOTE	

OBJECT	infGatewayTable
SYNTAX	SEQUENCE OF InfGatewayEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	(IP　　　　)
OID	{ infonetSystem 12 }
NOTE	

OBJECT	infGatewayEntry
SYNTAX	InfGatewayEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	(IP　　　　) 　　　　　　　　　　　　!

INDEX	infGatewayIndex
OID	{ infGatewayTable 1 }
NOTE	

OBJECT	infGatewayIndex
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	( ) IP , MIB 4 0 , IP
OID	{ infGatewayEntry 1 }
NOTE	

OBJECT	infGatewayIpAddress
SYNTAX	IpAddress
ACCESS	read-write
STATUS	mandatory
DSCRPT	IP IP 4 0 , MIB , IP , RIP , ( )
OID	{ infGatewayEntry 2 }
NOTE	

OBJECT	infGatewayStatus
SYNTAX	INTEGER { other(1), invalid(2), permanent(3) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	IP 1 : 2, 3 2 : ( ) 3 :
OID	{ infGatewayEntry 3 }
NOTE	

OBJECT	infBootSystem
SYNTAX	INTEGER { svFirm0(1), svFirm1(2) }
ACCESS	read-only

STATUS	mandatory
DSCRPT	
OID	{ infonetSystem 13 }
NOTE	

OBJECT	infChangeBootSystem
SYNTAX	INTEGER { keep(1), change(2) }
ACCESS	read-write
STATUS	mandatory
DSCRPT	2 set 2 set
OID	{ infonetSystem 14 }
NOTE	

OBJECT	infLinesetTable
SYNTAX	SEQUENCE OF InfLinesetEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	/
OID	{ infonetSystem 15 }
NOTE	

OBJECT	infLinesetEntry
SYNTAX	InfLinesetEntry
ACCESS	not-accessible
STATUS	mandatory
DSCRPT	/ !
INDEX	infLinesetSlot, infLinesetUnit
OID	{ infLinesetTable 1 }
NOTE	

OBJECT	infLinesetSlot
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	
OID	{ infLinesetEntry 1 }
NOTE	

OBJECT	infLinesetUnit
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	
OID	{ infLinesetEntry 2 }
NOTE	

OBJECT	infLinesetStatus
SYNTAX	INTEGER { other(1), up(2), failure(3), down(4) }
ACCESS	read-only
STATUS	mandatory
DSCRPT	/ 1 : 2 : 3 : 4 : halt
OID	{ infLinesetEntry 3 }
NOTE	

OBJECT	infWhyReset
SYNTAX	DisplayString
ACCESS	read-only
STATUS	mandatory
DSCRPT	
OID	{ infonetSystem 16 }
NOTE	

OBJECT	infSystemErrorPoint
SYNTAX	OCTET STRING (SIZE (4))
ACCESS	read-only
STATUS	mandatory

DSCRPT	,
	                                        infChIndex 8
	                  ,
	 0 :
	 1 :
	    , 1 ,
	 0(8 0) :
	 0 (8 1):
	         (1)

	MIB X'00600000' ,infChIndex 2 ,infChIndex 3
OID	{ infonetSystemError 1 }
NOTE	

OBJECT	infSystemErrorText1
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	mandatory
DSCRPT	MIB ,infSystemErrorText1,infSystemErrorText2 infSystemErrorText20 20 , 20 ,infSystemErrorText1 MIB , , , , 99 (DisplayString) ,
OID	{ infonetSystemError 2 }
NOTE	

OBJECT	infSystemErrorText2
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	mandatory
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 3 }
NOTE	

OBJECT	infSystemErrorText3
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	mandatory
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 4 }

NOTE	

OBJECT	infSystemErrorText4
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	mandatory
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 5 }
NOTE	

OBJECT	infSystemErrorText5
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	mandatory
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 6 }
NOTE	

OBJECT	infSystemErrorText6
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 7 }
NOTE	

OBJECT	infSystemErrorText7
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 8 }
NOTE	

OBJECT	infSystemErrorText8
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional

DSCRPT	
	infSystemErrorText1
OID	{ infonetSystemError 9 }
NOTE	

OBJECT	infSystemErrorText9
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	
	infSystemErrorText1
OID	{ infonetSystemError 10 }
NOTE	

OBJECT	infSystemErrorText10
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	
	infSystemErrorText1
OID	{ infonetSystemError 11 }
NOTE	

OBJECT	infSystemErrorText11
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	
	infSystemErrorText1
OID	{ infonetSystemError 12 }
NOTE	

OBJECT	infSystemErrorText12
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	
	infSystemErrorText1
OID	{ infonetSystemError 13 }
NOTE	

OBJECT	infSystemErrorText13

SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 14 }
NOTE	

OBJECT	infSystemErrorText14
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 15 }
NOTE	

OBJECT	infSystemErrorText15
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 16 }
NOTE	

OBJECT	infSystemErrorText16
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 17 }
NOTE	

OBJECT	infSystemErrorText17
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 18 }

NOTE	

OBJECT	infSystemErrorText18
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 19 }
NOTE	

OBJECT	infSystemErrorText19
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 20 }
NOTE	

OBJECT	infSystemErrorText20
SYNTAX	DisplayString (SIZE (0..255))
ACCESS	read-only
STATUS	optional
DSCRPT	infSystemErrorText1
OID	{ infonetSystemError 21 }
NOTE	

OBJECT	infCpuBusy5sec
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	5 CPU (%) boot 5 -1
OID	{ infonetCpuBusy 1 }
NOTE	

OBJECT	infCpuBusy1min
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	1 CPU (%) boot 5 -1

OID	{ infonetCpuBusy 2 }
NOTE	

OBJECT	infCpuBusy5min
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	5 CPU (%) boot 5 -1
OID	{ infonetCpuBusy 3 }
NOTE	

OBJECT	infMbufUsage
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	Mbuf (%)
OID	{ infonetMbufUsage 1 }
NOTE	

OBJECT	infMcbUsage
SYNTAX	INTEGER
ACCESS	read-only
STATUS	mandatory
DSCRPT	MCB (%)
OID	{ infonetMbufUsage 2 }
NOTE	

TRAP	infError
ENTERPRISE	infonetBase
DSCRPT	, MIB
SPECIFIC	1
NOTE	

TRAP	infPoff
ENTERPRISE	infonetBase
DSCRPT	,
SPECIFIC	2
NOTE	

3.17 FURUKAWA-INFMGTPROCESS-MIB

MODULE	infMgtProcs
UPDATED	200210030000Z
DSCRPT	MIB
REVISION	200210030000Z
DSCRPT	CPU
REVISION	200207170000Z
DSCRPT	Initial revision.
OID	{ infonetMgt 6 }
NOTE	

OBJECT	infMgtProcTable
SYNTAX	SEQUENCE OF InfMgtProcEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ infMgtProcs 1 }
NOTE	

OBJECT	infMgtProcEntry
SYNTAX	InfMgtProcEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
INDEX	infMgtProcIndex
OID	{ infMgtProcTable 1 }
NOTE	

OBJECT	infMgtProcIndex
SYNTAX	Integer32 (1..2147483647)
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	The auxiliary variable used for identifying instances of the columnar objects in the infMgtProcTable.
OID	{ infMgtProcEntry 1 }
NOTE	

OBJECT	infMgtProcPid
SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	PID
OID	{ infMgtProcEntry 2 }
NOTE	

OBJECT	infMgtProcName
SYNTAX	DisplayString(SIZE(1..255))
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtProcEntry 3 }
NOTE	

OBJECT	infMgtProcCPU5sec
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	5 CPU (%)
OID	{ infMgtProcEntry 4 }
NOTE	

OBJECT	infMgtProcCPU1min
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	1 CPU (%)
OID	{ infMgtProcEntry 5 }
NOTE	

OBJECT	infMgtProcCPU5min
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	5 CPU (%)
OID	{ infMgtProcEntry 6 }
NOTE	

OBJECT	infMgtProcRuntime
SYNTAX	Counter64

MAX-ACCESS	read-only
STATUS	current
DSCRPT	CPU
OID	{ infMgtProcEntry 7 }
NOTE	

OBJECT	infMgtProcCSW
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	CPU
OID	{ infMgtProcEntry 8 }
NOTE	

OBJECT	infMgtProcRtCSW
SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current
DSCRPT	1 CPU CPU
OID	{ infMgtProcEntry 9 }
NOTE	

OBJECT	infMgtProcPri
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtProcEntry 10 }
NOTE	

OBJECT	infMgtProcTT
SYNTAX	DisplayString(SIZE(1..255))
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtProcEntry 11 }
NOTE	

OBJECT	infMgtProcMEMReal
SYNTAX	Gauge32
MAX-ACCESS	read-only

STATUS	current
DSCRPT	(Byte)
OID	{ infMgtProcEntry 12 }
NOTE	

OBJECT	infMgtProcMEMVirt
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	(Byte)
OID	{ infMgtProcEntry 13 }
NOTE	

OBJECT	infMgtProcMEMPeak
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	(Byte)
OID	{ infMgtProcEntry 14 }
NOTE	

OBJECT	infMgtProcTotalCPU5sec
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	5 CPU (%)
OID	{ infMgtProcTotalCpu 1 }
NOTE	

OBJECT	infMgtProcTotalCPU1min
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	1 CPU (%)
OID	{ infMgtProcTotalCpu 2 }
NOTE	

OBJECT	infMgtProcTotalCPU5min
SYNTAX	Gauge32(0..100)
MAX-ACCESS	read-only
STATUS	current
DSCRPT	5 CPU (%)

OID	{ infMgtProcTotalCpu 3 }
NOTE	

OBJECT	infMgtProcTotalMEMTotal
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	(Byte)
OID	{ infMgtProcTotalMem 1 }
NOTE	

OBJECT	infMgtProcTotalMEMFree
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	(Byte)
OID	{ infMgtProcTotalMem 2 }
NOTE	

COMPLIANCE	infMgrProcCompliance
STATUS	current
DSCRPT	compliance statement
GROUP	infMgtProcGroup
DSCRPT	This group is mandatory for main-board.
GROUP	infMgtProcTotalCPUGroup
DSCRPT	This group is mandatory for main-board.
GROUP	infMgtProcTotalMemGroup
DSCRPT	This group is mandatory for main-board.
OID	{ infMgtProcCompliances 1}
NOTE	

OBJECT-G	infMgtProcGroup
OBJECTS	infMgtProcPid, infMgtProcName, infMgtProcCPU5sec, infMgtProcCPU1min, infMgtProcCPU5min, infMgtProcRuntime, infMgtProcCSW, infMgtProcRtCSW, infMgtProcPri, infMgtProcTT, infMgtProcMEMReal, infMgtProcMEMVirt, infMgtProcMEMPeak
STATUS	current
DSCRPT	
OID	{ infMgtProcGroups 1 }
NOTE	

OBJECT-G	infMgtProcTotalCPUGroup

OBJECTS	infMgtProcTotalCPU5sec, infMgtProcTotalCPU1min, infMgtProcTotalCPU5min
STATUS	current
DSCRPT	A collection of objects providing cpu on main-board.
OID	{ infMgtProcGroups 2 }
NOTE	

OBJECT-G	infMgtProcTotalMemGroup
OBJECTS	infMgtProcTotalMEMTotal, infMgtProcTotalMEMFree
STATUS	current
DSCRPT	A collection of objects providing memory on main-board.
OID	{ infMgtProcGroups 3 }
NOTE	

3.18 FURUKAWA-INFMGT-PATHCHECK-MIB

MODULE	infMgtPcheck
UPDATED	200401090000Z
DSCRPT	L3 MIB
REVISION	200401090000Z
DSCRPT	Initial revision.
OID	{ infonetMgt 14 }
NOTE	

OBJECT	infMgtPcheckLstNum
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckLst 1 }
NOTE	

OBJECT	infMgtPcheckLstTable
SYNTAX	SEQUENCE OF InfMgtPcheckLstEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ infMgtPcheckLst 2 }
NOTE	

OBJECT	infMgtPcheckLstEntry
SYNTAX	InfMgtPcheckLstEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
INDEX	infMgtPcheckLstIndex
OID	{ infMgtPcheckLstTable 1 }
NOTE	

OBJECT	infMgtPcheckLstIndex
SYNTAX	Integer32
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ infMgtPcheckLstEntry 1 }
NOTE	

OBJECT	infMgtPcheckLstDstAddrType
SYNTAX	InetAddressType
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckLstEntry 2 }
NOTE	

OBJECT	infMgtPcheckLstDstAddr
SYNTAX	InetAddress
MAX-ACCESS	read-only
STATUS	current
DSCRPT	infMgtPcheckLstDstAddrType
OID	{ infMgtPcheckLstEntry 3 }
NOTE	

OBJECT	infMgtPcheckLstNormalCond
SYNTAX	TruthValue
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckLstEntry 4 }
NOTE	

OBJECT	infMgtPcheckLstCurrentFails
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 5 }
NOTE	

OBJECT	infMgtPcheckLstCurrentSuccess
SYNTAX	Gauge32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 6 }
NOTE	

OBJECT	infMgtPcheckLstFailCond
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 7 }
NOTE	

OBJECT	infMgtPcheckLstSuccessCond
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 8 }
NOTE	

OBJECT	infMgtPcheckLstTrials
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping

OID	{ infMgtPcheckLstEntry 9 }
NOTE	

OBJECT	infMgtPcheckLstInterval
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 10 }
NOTE	

OBJECT	infMgtPcheckLstReducedInterval
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	ping
OID	{ infMgtPcheckLstEntry 11 }
NOTE	

OBJECT	infMgtPcheckLstLastChange
SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current
DSCRPT	infMgtPcheckLstNormalCond
OID	{ infMgtPcheckLstEntry 12 }
NOTE	

OBJECT	infMgtPcheckFilNum
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckFil 1 }
NOTE	

OBJECT	infMgtPcheckFilTable
SYNTAX	SEQUENCE OF InfMgtPcheckFilEntry
MAX-ACCESS	not-accessible
STATUS	current

DSCRPT	
OID	{ infMgtPcheckFil 2 }
NOTE	

OBJECT	infMgtPcheckFilEntry
SYNTAX	InfMgtPcheckFilEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
INDEX	infMgtPcheckFilIndex
OID	{ infMgtPcheckFilTable 1 }
NOTE	

OBJECT	infMgtPcheckFilIndex
SYNTAX	Integer32
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ infMgtPcheckFilEntry 1 }
NOTE	

OBJECT	infMgtPcheckFilPListIndex
SYNTAX	Integer32
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckFilEntry 2 }
NOTE	

OBJECT	infMgtPcheckFilDstAddrType
SYNTAX	InetAddressType
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckFilEntry 3 }
NOTE	

OBJECT	infMgtPcheckFilDstAddr
SYNTAX	InetAddress
MAX-ACCESS	read-only

STATUS	current
DSCRPT	infMgtPcheckFilDstAddrType
OID	{ infMgtPcheckFilEntry 4 }
NOTE	

OBJECT	infMgtPcheckFilDstPfxLen
SYNTAX	InetAddressPrefixLength
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ infMgtPcheckFilEntry 5 }
NOTE	

OBJECT	infMgtPcheckFilNextIfIndex1
SYNTAX	InterfaceIndex
MAX-ACCESS	read-only
STATUS	current
DSCRPT	1st ifIndex
OID	{ infMgtPcheckFilEntry 6 }
NOTE	

OBJECT	infMgtPcheckFilNextIfName1
SYNTAX	DisplayString
MAX-ACCESS	read-only
STATUS	current
DSCRPT	1st ifDescr
OID	{ infMgtPcheckFilEntry 7 }
NOTE	

OBJECT	infMgtPcheckFilNextIfIndex2
SYNTAX	InterfaceIndexOrZero
MAX-ACCESS	read-only
STATUS	current
DSCRPT	2nd ifIndex 0

OID	{ infMgtPcheckFilEntry 8 }
NOTE	

OBJECT	infMgtPcheckFilNextIfName2
SYNTAX	DisplayString
MAX-ACCESS	read-only
STATUS	current
DSCRPT	2nd ifDescr
OID	{ infMgtPcheckFilEntry 9 }
NOTE	

COMPLIANCE	infMgtPcheckConfCompliance
STATUS	current
DSCRPT	compliance statement
MANDATORY-G	infMgtPcheckLstGroup, infMgtPcheckFilGroup
OID	{ infMgtPcheckConfCompliances 1 }
NOTE	

OBJECT-G	infMgtPcheckLstGroup
OBJECTS	infMgtPcheckLstNum, infMgtPcheckLstDstAddrType, infMgtPcheckLstDstAddr, infMgtPcheckLstNormalCond, infMgtPcheckLstCurrentFails, infMgtPcheckLstCurrentSuccess, infMgtPcheckLstFailCond, infMgtPcheckLstSuccessCond, infMgtPcheckLstTrials, infMgtPcheckLstInterval, infMgtPcheckLstReducedInterval, infMgtPcheckLstLastChange
STATUS	current
DSCRPT	A Collection of objects providing path check list.
OID	{ infMgtPcheckConfGroups 1 }
NOTE	

OBJECT-G	infMgtPcheckFilGroup
OBJECTS	infMgtPcheckFilNum, infMgtPcheckFilPListIndex, infMgtPcheckFilDstAddrType, infMgtPcheckFilDstAddr, infMgtPcheckFilDstPfxLen, infMgtPcheckFilNextIfIndex1, infMgtPcheckFilNextIfName1, infMgtPcheckFilNextIfIndex2, infMgtPcheckFilNextIfName2
STATUS	current
DSCRPT	A Collection of objects providing path filter list.
OID	{ infMgtPcheckConfGroups 2 }
NOTE	

3.19 FURUKAWA-INFMGT-SYSTEM-RESOURCES-MIB

MODULE	infMgtSystemResourcesMIB
UPDATED	200412070000Z
DSCRPT	System Resources MIBs. MIB
REVISION	200412070000Z
DSCRPT	Initial revision.
OID	{ infonetMgt 17 }
NOTE	

TC	InfMgtSysResrcIndex
STATUS	current
DSCRPT	
SYNTAX	Integer32 (1..2147483647)

TC	InfMgtSysResrcString
STATUS	current
DSCRPT	DisplayString (RFC 2579)
SYNTAX	OCTET STRING (SIZE (1..255))

TC	InfMgtSysResrcGauge
STATUS	current
DSCRPT	Gauge32(RFC2578)
SYNTAX	Integer32 (-1..2147483647)

TC	InfMgtSysResrcCounter
STATUS	current
DSCRPT	Counter32(RFC2578)
SYNTAX	Integer32 (-1..2147483647)

OBJECT	inSysResrcTable
SYNTAX	SEQUENCE OF InSysResrcEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ infMgtSysResrcMIBObjects 1 }
NOTE	

OBJECT	inSysResrcEntry
SYNTAX	InSysResrcEntry
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
INDEX	inSysResrcIndex
OID	{ inSysResrcTable 1 }
NOTE	

OBJECT	inSysResrcIndex
SYNTAX	InfMgtSysResrcIndex
MAX-ACCESS	not-accessible
STATUS	current
DSCRPT	
OID	{ inSysResrcEntry 1 }
NOTE	1 2

OBJECT	inSysResrcName
SYNTAX	InfMgtSysResrcString
MAX-ACCESS	read-only
STATUS	current
DSCRPT	
OID	{ inSysResrcEntry 2 }
NOTE	index Name 1 NAT Sessions 2 Stateful Inspection Sessions

OBJECT	inSysResrcMaxEntries
SYNTAX	Integer32 (-1..2147483647)
UNITS	Entries
MAX-ACCESS	read-only
STATUS	current
DSCRPT	-1
OID	{ inSysResrcEntry 3 }
NOTE	

OBJECT	inSysResrcReservedEntries

SYNTAX	Integer32 (-1..2147483647)
UNITS	Entries
MAX-ACCESS	read-only
STATUS	current
DSCRPT	-1
OID	{ inSysResrcEntry 4 }
NOTE	Stateful Inspection Sessions ”-1”

OBJECT	inSysResrcActiveEntries
SYNTAX	InfMgtSysResrcGauge
UNITS	Entries
MAX-ACCESS	read-only
STATUS	current
DSCRPT	-1
OID	{ inSysResrcEntry 5 }
NOTE	

OBJECT	inSysResrcPeakEntries
SYNTAX	InfMgtSysResrcGauge
UNITS	Entries
MAX-ACCESS	read-only
STATUS	current
DSCRPT	-1
OID	{ inSysResrcEntry 6 }
NOTE	”-1”

OBJECT	inSysResrcLimitFails
SYNTAX	InfMgtSysResrcCounter
UNITS	Failures
MAX-ACCESS	read-only
STATUS	current
DSCRPT	-1
OID	{ inSysResrcEntry 7 }
NOTE	”-1”

COMPLIANCE	infMgtSysResrcMIBCompliance
STATUS	current
DSCRPT	SystemResources MIB compliance statement
MANDATORY-G	inSysResrcGroup
OID	{ infMgtSysResrcMIBCompliances 1 }
NOTE	

OBJECT-G	inSysResrcGroup
OBJECTS	inSysResrcIndex, inSysResrcName, inSysResrcMaxEntries, inSysResrcReservedEntries, inSysResrcActiveEntries, inSysResrcPeakEntries, inSysResrcLimitFails
STATUS	current
DSCRPT	
OID	{ infMgtSysResrcMIBGroups 1 }
NOTE	

Management Information Base for Network Management of TCP/IP-based internets: MIB-II　Mar/1991　IETF　RFC1213

A Convention for Defining Traps for use with the SNMP　Mar/1991　IETF　RFC1215

Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)　Jan/1996　IETF　RFC1907

SNMPv2 Management Information Base for the Internet Protocol using SMIv2　Nov/1996　IETF　RFC2011

SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2　Nov/1996　IETF　RFC2012

SNMPv2 Management Information Base for the User DatagramProtocol using SMIv2　Nov/1996　IETF　RFC2013

IP Forwarding Table MIB　Jan/1997　IETF　RFC2096

Management Information Base for IP Version 6: Textual Conventions and General Group　Dec/1998　IETF　RFC2465

Management Information Base for IP Version 6: ICMPv6 Group　Dec/1998　IETF　RFC2466

An Architecture for Describing SNMP Management Frameworks　Apr/1999　IETF　RFC2571

Structure of Management Information Version 2 (SMIv2)　Apr/1999　IETF　RFC2578

Textual Conventions for SMIv2　Apr/1999　IETF　RFC2579

The Interfaces Group MIB　Jun/2000　IETF　RFC2863

Textual Conventions for Internet Network Addresses　May/2002　IETF　RFC3291