Chapter B

SNMP MIBs

Introduction

This appendix describes the *Management Information Bases (MIBs)* and managed objects implemented by the AR400 router, including any variations from the RFC standards. Table B-1 lists the MIBs supported by the router. For information about configuring SNMP on the router see Chapter 38, Simple Network Management Protocol (SNMP).

Table B-1: MIBs supported by the router

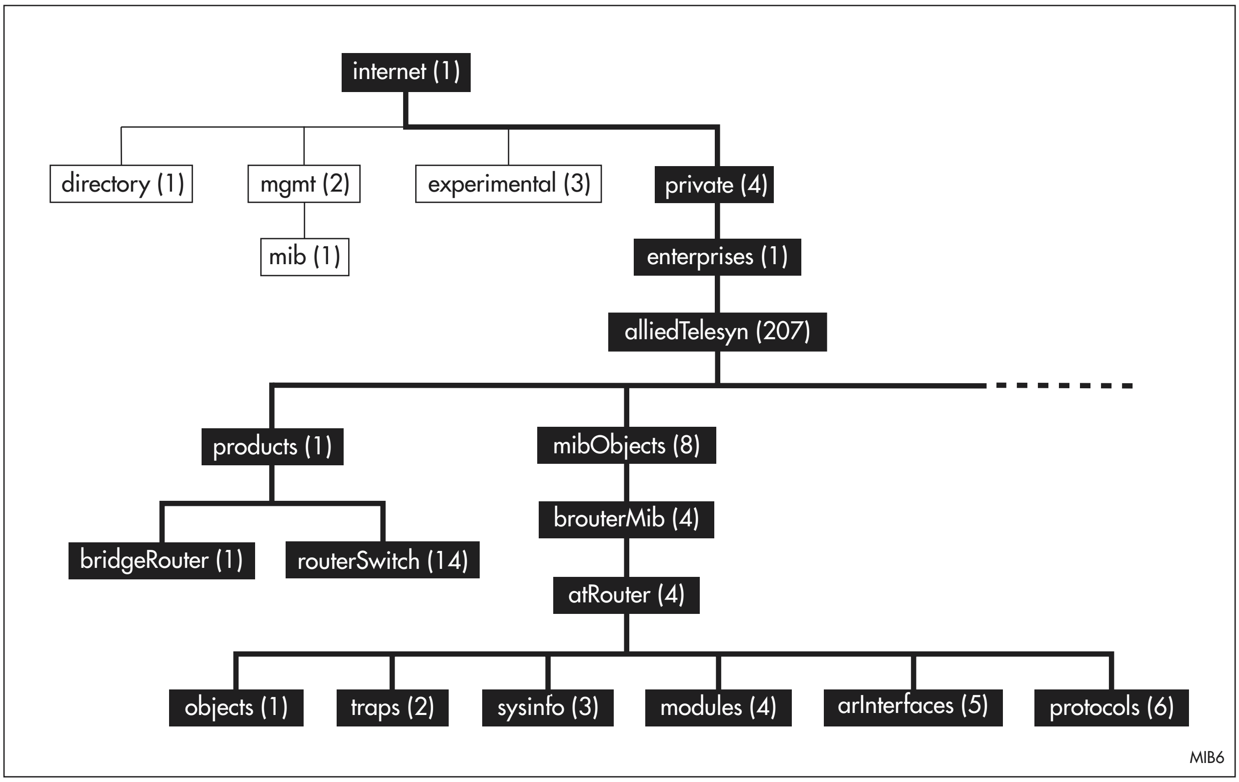
SNMP MIB	Description
Allied Telesyn Enterprise MIB	Objects in the Allied Telesyn Enterprise MIB for managing the router.
MIB-II	The core set of objects for TCP/IP internets.
IP Forwarding Table MIB	Objects for managing Classless Inter-Domain Routing (CIDR).
Ethernet-like Interface Types MIB	Objects for managing Ethernet-like interfaces.
Bridge MIB	Objects for managing MAC bridges based on the IEEE 802.1D-1990 standard.
ATM MIB	Objects for managing ATM interfaces.
ADSL Line MIB	Objects for managing ADSL interfaces.
Port Access Control MIB	Objects for managing port access control based on the IEEE 802.1X standard.
RMON MIB	Objects for managing remote network monitoring devices.
Frame Relay DTE MIB (PIC bay only)	Objects for managing Frame Relay.
DS1, E1, DS2 and E2 Interface Types MIB	Objects for managing DS1, E1, DS2 and E2 interfaces.
Host Resources MIB	Objects for managing host systems.
SNMP v3 MIB	Objects for managing SNMP v3.
Border Gateway Protocol v4 (BGP-4) MIB	Objects for managing the Border Gateway Protocol version 4 (BGP-4).

Allied Telesyn Enterprise MIB

The *Allied Telesyn Enterprise MIB* defines a portion of the Management Information Base (MIB) for managing Allied Telesyn products. Only that portion of the Allied Telesyn Enterprise MIB relating to the AR400 router is described here. The AR400 router supports a number of standard MIBs. Objects in the Enterprise MIB represent features specific to the AR400 router that are not covered by other standard MIBs.

Objects defined in this MIB reside in the private(4) sub-tree (Figure B-1 on page B-3) and have the object identifier prefix *alliedTelesyn* ({ enterprises 207 }).

Figure B-1: The Allied Telesyn Enterprise MIB sub-tree of the Internet-standard Management Information Base (MIB)



The Products Sub-tree

The Products sub-tree contains a set of object identifiers for Allied Telesyn products. Objects have the identifier prefix of *products* ({ alliedTelesyn 1 }). Within this sub-tree, objects with the identifier prefix *bridgeRouter* ({ products 1 }) are routers, and objects with the identifier prefix *routerSwitch* ({ products 14 }) are switches with routing functionality (Table B-2 on page B-4).

Table B-2: Object identifiers for Allied Telesyn router products

Object	Object Identifier	Description
bridgeRouter	{ products 1 }	Router products
centreCOM-AR300Router	{ bridgeRouter 8 }	AT-AR300 Basic Rate ISDN (S/T interface) Router, 2 voice ports
centreCOM-AR720Router	{ bridgeRouter 11 }	AT-AR720 Modular Network Access Platform
centreCOM-AR300LRouter	{ bridgeRouter 12 }	AT-AR300L Basic Rate ISDN (S/T interface) Router
centreCOM-AR310Router	{ bridgeRouter 13 }	AT-AR310 Basic Rate ISDN (S/T interface) Router, 4 voice ports
centreCOM-AR300LURouter	{ bridgeRouter 14 }	AT-AR300LU Basic Rate ISDN (U interface) Router
centreCOM-AR300URouter	{ bridgeRouter 15 }	AT-AR300U Basic Rate ISDN (U interface) Router, 2 voice ports
centreCOM-AR310URouter	{ bridgeRouter 16 }	AT-AR310U Basic Rate ISDN (U interface) Router, 4 voice ports
centreCOM-AR350Router	{ bridgeRouter 17 }	AT-AR350 Synchronous Router
centreCOM-AR370Router	{ bridgeRouter 18 }	AT-AR370 Synchronous/Basic Rate ISDN (S/T interface) Router
centreCOM-AR330Router	{ bridgeRouter 19 }	AT-AR330 Synchronous/Dual Ethernet Router
centreCOM-AR395Router	{ bridgeRouter 20 }	AT-AR395 E1/Primary Rate ISDN Router
centreCOM-AR390Router	{ bridgeRouter 21 }	AT-AR390 E1/G.703 Unchannelised ISDN Router
centreCOM-AR370URouter	{ bridgeRouter 22 }	AT-AR370U Synchronous/Basic Rate ISDN (U interface) Router
centreCOM-AR740Router	{ bridgeRouter 23 }	AT-AR740 Modular Branch Office Router
centreCOM-AR140SRouter	{ bridgeRouter 24 }	AT-AR140 Internet Access Router, Basic Rate ISDN (S/T interface), 2 voice ports
centreCOM-AR140URouter	{ bridgeRouter 25 }	AT-AR140 Internet Access Router, Basic Rate ISDN (U interface), 2 voice ports
centreCOM-AR320Router	{ bridgeRouter 26 }	AT-AR320 Dual Ethernet Firewall Router
centreCOM-AR130SRouter	{ bridgeRouter 27 }	AT-AR130 Internet Access Router, Basic Rate ISDN (S/T interface)
centreCOM-AR130URouter	{ bridgeRouter 28 }	AT-AR130 Internet Access Router, Basic Rate ISDN (U interface)
centreCOM-AR160Router	{ bridgeRouter 29 }	AT-AR160 Internet Access Router
at-AR740RouterDC	{ bridgeRouter 43 }	AT-AR740 Modular Branch Office Router (DC power)
centreCOM-AR120Router	{ bridgeRouter 44 }	AT-AR120 Internet Access Router
at-AR410Router	{ bridgeRouter 47 }	AT-AR410 Branch Office Router
at-AR725Router	{ bridgeRouter 48 }	AT-AR725 Modular Network Access Platform
at-AR745Router	{ bridgeRouter 49 }	AT-AR745 Modular Branch Office Router
at-AR410v2Router	{ bridgeRouter 50 }	AT-AR410v2 Branch Office Router (version 2)
at-AR410v3Router	{ bridgeRouter 51 }	AT-AR410v3 Branch Office Router (version 3)
at-AR725RouterDC	{ bridgeRouter 52 }	AT-AR725 Modular Branch Office Route (DC power)
at-AR745RouterDC	{ bridgeRouter 53 }	AT-AR745 Modular Branch Office Route (DC power)
at-AR450Router	{ bridgeRouter 54 }	CentreCOM-AT-AR450S Broadband Router
at-AR450DualRouter	{ bridgeRouter 55 }	AT-AR450S Security Appliance
routerSwitch	{ products 14 }	Switch products
at-Rapier24	{ routerSwitch 1 }	Rapier 24 switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-Rapier16fSC	{ routerSwitch 2 }	Rapier 16F-FX/SC switch, 16 x 100BASE-FX ports (SC fibre connectors)
at-rapier16fVF	{ routerSwitch 3 }	*—Deprecated—*
at-Rapier16fMT	{ routerSwitch 4 }	Rapier 16F-FX/MT-RJ switch, 16 x 100BASE-FX ports (MT-RJ fibre connectors)

Table B-2: Object identifiers for Allied Telesyn router products (continued)

Object	Object Identifier	Description
at-Rapier48	{ routerSwitch 5 }	Rapier 48 switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-Rapier8t8fSC	{ routerSwitch 6 }	Rapier 8/8SC switch, 8 x 10BASE-T/100BASE-TX ports (RJ-45 connectors), 8 x 100BASE-FX ports (SC fibre connectors)
at-Rapier8t8fSCi	{ routerSwitch 7 }	*—Reserved—*
at-Rapier8t8fMT	{ routerSwitch 8 }	Rapier 8/8MT switch, 8 x 10BASE-T/100BASE-TX ports (RJ-45 connectors), 8 x 100BASE-FX ports (MT-RJ fibre connectors)
at-Rapier8t8fMTi	{ routerSwitch 9 }	*—Reserved—*
at-Rapier8fSC	{ routerSwitch 10 }	Rapier 8SC switch, 8 x 100BASE-FX ports (SC fibre connectors)
at-Rapier8fSCi	{ routerSwitch 11 }	*—Reserved—*
at-Rapier8fMT	{ routerSwitch 12 }	Rapier 8MT switch, 8 x 100BASE-FX ports (MT-RJ fibre connectors)
at-Rapier8fMTi	{ routerSwitch 13 }	*—Reserved—*
at-Rapier16fMTi	{ routerSwitch 14 }	*—Reserved—*
at-RapierG6	{ routerSwitch 15 }	Rapier G6 switch, 6 x 100BASE-TX/1000BASE-T ports (RJ-45 connectors)
at-RapierG6SX	{ routerSwitch 16 }	Rapier G6F-SX/SC switch, 6 x 1000BASE-SX ports (SC fibre connectors)
at-RapierG6LX	{ routerSwitch 17 }	Rapier G6F-LX/SC switch, 6 x 1000BASE-LX ports (SC fibre connectors)
at-RapierG6MT	{ routerSwitch 18 }	Rapier G6F-SX/MT-RJ switch, 6 x 1000BASE-SX ports (MT-RJ fibre connectors)
at-Rapier16fSCi	{ routerSwitch 19 }	Rapier 16F-FX/SC*i* switch, 16 x 100BASE-FX ports (SC fibre connectors)
at-Rapier24i	{ routerSwitch 20 }	Rapier 24*i* switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-Rapier48i	{ routerSwitch 21 }	Rapier 48*i* switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-Switchblade4AC	{ routerSwitch 22 }	AT-SB4104 SwitchBlade 4 Chassis (AC power)
at-Switchblade4DC	{ routerSwitch 23 }	AT-SB4104 SwitchBlade 4 Chassis (DC power)
at-Switchblade8AC	{ routerSwitch 24 }	AT-SB4108 SwitchBlade 8 Chassis (AC power)
at-Switchblade8DC	{ routerSwitch 25 }	AT-SB4108 SwitchBlade 8 Chassis (DC power)
at-9816GF	{ routerSwitch 26 }	AT-9816GF Gigabit switch, 16 x 1000BASE-X ports (GBIC slots)
at-9812TF	{ routerSwitch 27 }	AT-9812TF Gigabit switch, 12 x 10BASE-T/100BASE-TX/1000BASE-T ports (RJ-45 connectors), 4 x 1000BASE-X ports (GBIC slots)
at-9816GB	{ routerSwitch 28 }	AT-9816GB Gigabit switch, 16 x 1000BASE-X ports (GBIC slots)
at-9812T	{ routerSwitch 29 }	AT-9812T Gigabit switch, 12 x 10BASE-T/100BASE-TX/1000BASE-T ports (RJ-45 connectors), 4 x 1000BASE-X ports (GBIC slots)
at-8724XL	{ routerSwitch 30 }	AT-8724 switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-8748XL	{ routerSwitch 31 }	AT-8748XL switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-8724XLDC	{ routerSwitch 32 }	AT-8724XL switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
at-8748XLDC	{ routerSwitch 33 }	AT-8748 switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)

Table B-2: Object identifiers for Allied Telesyn router products (continued)

Object	Object Identifier	Description
at-9816GB-DC	{ routerSwitch 34 }	AT-9816GB Gigabit switch, 16 x 1000BASE-X ports (GBIC slots) (DC power)
at-9812T-DC	{ routerSwitch 35 }	AT-9812T Gigabit switch, 12 x 10BASE-T/100BASE-TX/1000BASE-T ports (RJ-45 connectors), 4 x 1000BASE-X ports (GBIC slots) (DC power)
at-8824	{ routerSwitch 36 }	AT-8824 switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2-port 1000BASE-X (GBIC slots)
at-8848	{ routerSwitch 37 }	AT-8848 switch, 48 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2-port 1000BASE-X (GBIC slots)
at-8824-DC	{ routerSwitch 38 }	AT-8824 switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2-port 1000BASE-X (GBIC slots) (DC power)
at-8848-DC	{ routerSwitch 39 }	AT-8848 switch, 48 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2-port 1000BASE-X (GBIC slots) (DC power)
at-8624XL-80	{ routerSwitch 41 }	CentreCOM 8624XL switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
at-8724XL-80	{ routerSwitch 42 }	CentreCOM 8724XL switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
at-8748XL-80	{ routerSwitch 43 }	CentreCOM 8748XL switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
at-8948EX	{ routerSwitch 44 }	AT-8948 Multi-layer Fast Ethernet Switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors), 4 SFP Gigabit uplink ports
at-8624T2M	{ routerSwitch 46 }	AT-8624T2M Layer 3 Fast Ethernet Switch
at-Rapier24i-DC-NEBS	{ routerSwitch 47 }	Rapier 24*i* switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power) NEBS compliant
at-8724XL-DC-NEBS	{ routerSwitch 48 }	AT-8724XL switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power) NEBS compliant
at-9924T	{ routerSwitch 49 }	AT-9924T Multi-layer IPv4 Gigabit Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 4 SFP Gigabit uplink ports
at-9924SP	{ routerSwitch 50 }	AT-9924SP Multi-layer IPv4 Gigabit Switch, 24 SFP Gigabit uplink ports
at-9924T-4SP	{ routerSwitch 51 }	AT-9924T/4SP Multi-layer IPv4 and IPv6 Gigabit Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 4 SFP Gigabit uplink ports

The AT Router Sub-tree

The AT Router sub-tree contains a set of objects for managing multiprotocol routers and switches. Objects have the identifier prefix of *atRouter* ({ alliedTelesyn mibObject brouterMib 4 }). Objects are arranged into six groups (Table B-3 on page B-6).

Table B-3: Object groups in the AT Router sub-tree of the Allied Telesyn Enterprise MIB

Group	Object Identifier (OID)
mibObject	{ alliedTelesyn 8 }
brouterMib	{ mibObject 4 }

Table B-3: Object groups in the AT Router sub-tree of the Allied Telesyn Enterprise MIB

Group	Object Identifier (OID)
atRouter	{ brouterMib 4 }
objects	{ atRouter 1 }
traps	{ atRouter 2 }
sysinfo	{ atRouter 3 }
modules	{ atRouter 4 }
arInterfaces	{ atRouter 5 }
protocols	{ atRouter 6 }

The Objects Group

The Objects Group contains four sets of object identifiers for boards (Table B-4 on page B-7), releases ({ objects 2 }), interface types (Table B-5 on page B-11) and chip sets (Table B-6 on page B-12). These object identifiers are for use with the *hrDeviceID* object in the Host Resources MIB (Table B-11 on page B-44).

Table B-4: Object identifiers for base CPU and expansion boards

Object	Object Identifier	Description
boards	{ objects 1 }	
pprIcmAr023	{ boards 39 }	AT-AR023 SYN1 PIC card
pprIcmAr021s	{ boards 40 }	AT-AR021s BRI1 (S/T interface) PIC card
pprIcmAr022	{ boards 41 }	AT-AR022 ETH1 PIC card
pprIcmAr025	{ boards 45 }	AT-AR025 E1 PRI1 PIC card
pprIcmAr024	{ boards 46 }	AT-AR024 ASYN4 PIC card
pprAr300	{ boards 49 }	AT-AR300 (1 BRI S/T ISDN port, 2 POTS ports)
pprAr300L	{ boards 52 }	AT-AR300L (1 BRI S/T ISDN port)
pprAr310	{ boards 53 }	AT-AR310 (1 BRI S/T ISDN port, 4 POTS ports)
pprAr120	{ boards 54 }	AT-AR120 (4 LAN ports, 1 PCMCIA slot)
pprAr300Lu	{ boards 55 }	AT-AR300LU (1 BRI U ISDN port)
pprAr300u	{ boards 56 }	AT-AR300U (1 BRI U ISDN port, 2 POTS ports)
pprAr310u	{ boards 57 }	AT-AR310U (1 BRI U ISDN port, 4 POTS ports)
pprAr350	{ boards 58 }	AT-AR350 (1 Sync port)
pprIcmAr021u	{ boards 59 }	AT-AR021u BRI1 (U interface) PIC card
pprAr720	{ boards 63 }	AT-AR720 (2 PIC card slots)
pprAr010	{ boards 67 }	AT-AR010 EMAC Encryption MAC card
pprAr012	{ boards 68 }	AT-AR012 CMAC Compression MAC card
pprAr011	{ boards 69 }	AT-AR011 CEMAC Encryption/Compression MAC card
pprAr370	{ boards 70 }	AT-AR370 (1 Sync port, 1 BRI S/T ISDN port)
pprAr330	{ boards 71 }	AT-AR330 (1 Sync port, 2 LAN ports)
pprAr395	{ boards 72 }	AT-AR395 (1 G.703/PRI ISDN port)
pprAr390	{ boards 73 }	AT-AR390 (1 G.703/PRI unchannelised port)
pprAr370u	{ boards 75 }	AT-AR370U (1 Sync port, 1 BRI U ISDN port)
pprIcmAr020	{ boards 76 }	AT-AR020 T1 PRI1 PIC card

Table B-4: Object identifiers for base CPU and expansion boards (continued)

Object	Object Identifier	Description
pprAr740	{ boards 79 }	AT-AR740 (2 10/100 LAN ports, 2 RS-232 asynchronous ports, 1 NSM expansion bay, 2 PIC expansion bays)
pprAr140s	{ boards 80 }	AT-AR140 (1 BRI S/T ISDN port, 2 voice ports)
pprAr140u	{ boards 81 }	AT-AR140 (1 BRI U ISDN port, 2 voice ports)
pprAr160su	{ boards 82 }	AT-AR160 (2 daisy-chained BRI S/T ISDN ports, 1 BRI U ISDN port, 2 voice ports)
pprAr320	{ boards 83 }	AT-AR320 (2 LAN ports)
pprAr130s	{ boards 85 }	AT-AR130 (1 BRI S/T ISDN port)
pprAr130u	{ boards 86 }	AT-AR130 (1 BRI U ISDN port)
pprRapier24	{ boards 87 }	Rapier 24 (24 10BASE-T/100BASE-TX ports)
pprNsm0404Pic	{ boards 88 }	AT-AR040 NSM (4 PIC expansion bays)
pprA35SXSC	{ boards 89 }	AT-A35/SX uplink module (1 1000BASE-SX, SC connector)
pprA35LXSC	{ boards 90 }	AT-A35/LX uplink module (1 1000BASE-LX, SC connector)
pprA36MTRJ	{ boards 91 }	*—Reserved—*
pprA37VF45	{ boards 92 }	*—Reserved—*
pprA38LC	{ boards 93 }	*—Reserved—*
pprA39Tx	{ boards 94 }	AT-A39 uplink module (1 10/100/1000BASE-T, RJ-45 connector)
pprAr740DC	{ boards 95 }	AT-AR740 (2 10/100 LAN ports, 2 RS-232 async ports, 1 NSM expansion bay, 2 PIC expansion bays, DC power)
pprNsm0418BRI	{ boards 96 }	AT-AR041 NSM (8 BRI S/T ISDN ports)
pprRapier16fSC	{ boards 97 }	Rapier 16f/SC (16 100BASE-FX ports, SC fibre connectors)
ppr8624xl80	{ boards 98 }	AT-8424XL (24 10BASE-T/100BASE-T ports, RJ-45 connectors)
pprRapier16fMT	{ boards 99 }	Rapier 16f/MT-RJ (16 100BASE-FX ports, MT-RJ fibre connectors)
pprRapier16fMTi	{ boards 100 }	Rapier 16fi/MT-RJ (16 100BASE-FX ports, MT-RJ fibre connectors)
pprRapier8t8fSC	{ boards 101 }	Rapier 8/8SC (8 10BASE-T/100BASE-T ports, RJ-45 connectors; 8 100BASE-F ports, SC fibre connectors)
pprRapier8t8fSCi	{ boards 102 }	*—Reserved—*
pprRapier8t8fMT	{ boards 103 }	Rapier 8/8MT (8 10BASE-T/100BASE-T ports, RJ-45 connectors; 8 100BASE-F ports, MT-RJ fibre connectors)
pprRapier8t8fMTi	{ boards 104 }	*—Reserved—*
pprRapier8fSC	{ boards 105 }	Rapier 8SC (8 100BASE-F ports, SC fibre connectors)
pprRapier8fSCi	{ boards 106 }	*—Reserved—*
pprRapier8fMT	{ boards 107 }	Rapier 8MT (8 100BASE-F ports, MT-RJ fibre connectors)
pprRapier8fMTi	{ boards 108 }	*—Reserved—*
pprRapierG6	{ boards 110 }	Rapier G6 (6 100BASE-TX/1000BASE-T ports, RJ-45 connectors)
pprRapierG6SX	{ boards 111 }	Rapier G6-SX/SC (6 1000BASE-S ports, SC fibre connectors)
pprRapierG6LX	{ boards 112 }	Rapier G6f-LX/SC (6 1000BASE-L ports, SC fibre connectors)
pprRapierG6MT	{ boards 113 }	Rapier G6f-SX/MT-RJ (6 1000BASE-S ports, MT-RJ fibre connectors)
pprRapier16fSCi	{ boards 114 }	Rapier 16fi/SC (16 100BASE-F ports, SC fibre connectors)
pprRapier24i	{ boards 115 }	Rapier 24i (24 10BASE-T/100BASE-T ports, RJ-45 connectors)
pprAr824	{ boards 116 }	AR824 (24 10BASE-T/100BASE-T ports, RJ-45 connectors)

Table B-4: Object identifiers for base CPU and expansion boards (continued)

Object	Object Identifier	Description
pprAr816fSC	{ boards 117 }	AR816f/SC (16 100BASE-FX ports, SC fibre connectors)
pprAr816fSCi	{ boards 118 }	*—Reserved—*
pprAr816fMT	{ boards 119 }	AR816f/MT-RJ (16 100BASE-FX ports, MT-RJ fibre connectors)
pprAr816fMTi	{ boards 120 }	*—Reserved—*
pprAr88t8fSC	{ boards 121 }	*—Reserved—*
pprAr88t8fSCi	{ boards 122 }	*—Reserved—*
pprAr88t8fMT	{ boards 123 }	*—Reserved—*
pprAr88t8fMTi	{ boards 124 }	*—Reserved—*
pprAr88fSC	{ boards 125 }	*—Reserved—*
pprAr88fSCi	{ boards 126 }	*—Reserved—*
pprAr88fMT	{ boards 127 }	*—Reserved—*
pprAr88fMTi	{ boards 128 }	*—Reserved—*
pprAr824i	{ boards 129 }	*—Reserved—*
pprAt8724XL	{ boards 130 }	AT-8724XL (24 10BASE-T/100BASE-TX ports, RJ-45 connectors)
pprAt8748XL	{ boards 131 }	AT-8748XL (48 10BASE-T/100BASE-TX ports, RJ-45 connectors)
pprAt8724XLDC	{ boards 132 }	AT-8724XL (24 10BASE-T/100BASE-TX ports, RJ-45 connectors, DC power)
pprAt8748XLDC	{ boards 133 }	AT-8748XL (48 10BASE-T/100BASE-TX ports, RJ-45 connectors, DC power)
pprAt8824	{ boards 134 }	AT-8824 switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors)
pprAt8824DC	{ boards 135 }	AT-8824 switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
pprAt8724XLDC	{ boards 141 }	CentreCOM 8724XL switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
pprAt8748XLDC	{ boards 142 }	CentreCOM 8748XL switch, 48 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
pprRapier24iDC-NEBS	{ boards 144 }	Rapier 24*i* switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power) NEBS compliant
pprAt8724XLDC-NEBS	{ boards 146 }	AT-8724XL switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power) NEBS compliant
pprAt8848DC	{ boards 147 }	AT-8848 switch, 48 10BASE-T/100BASE-TX ports (RJ-45 connectors) (DC power)
pprRapier48	{ boards 148 }	Rapier 48 (48 10BASE-T/100BASE-T ports, RJ-45 connectors)
pprAt8848	{ boards 149 }	AT-8848 switch, 48 10BASE-T/100BASE-TX ports (RJ-45 connectors)
pprRapier48i	{ boards 150 }	Rapier 48i (48 10BASE-T/100BASE-T ports, RJ-45 connectors)
pprNsm0424BRI	{ boards 151 }	AT-AR042 NSM (4 BRI S/T ISDN ports)
pprIcmAR026	{ boards 153 }	AT-AR026 4ETH PIC (4 10-BaseT/100Base-TX LAN ports)
ppr9816GF	{ boards 157 }	AT-9816GF Gigabit switch, 16 x 1000BASE-X ports (GBIC slots)
ppr9812TF	{ boards 158 }	AT-9812TF Gigabit switch, 12 x 10BASE-T/100BASE-TX/ 1000BASE-T ports (RJ-45 connectors), 4 x 1000BASE-X ports (GBIC slots)
pprSbChassis4AC	{ boards 159 }	AT-SB4104 SwitchBlade 4 Chassis (AC power)

Table B-4: Object identifiers for base CPU and expansion boards (continued)

Object	Object Identifier	Description
pprSbChassis4DC	{ boards 160 }	AT-SB4104 SwitchBlade 4 Chassis (DC power)
pprSbChassis8AC	{ boards 161 }	AT-SB4108 SwitchBlade 8 Chassis (AC power)
pprSbChassis8DC	{ boards 162 }	AT-SB4108 SwitchBlade 8 Chassis (DC power)
pprSbChassis16AC	{ boards 163 }	*—Reserved—*
pprSbChassis16DC	{ boards 164 }	*—Reserved—*
pprSbControl	{ boards 165 }	AT-SB4211 Switch Controller
pprSbControlDTM	{ boards 166 }	*—Reserved—*
pprSb48t	{ boards 167 }	AT-SB4311 48-Port (RJ-45) Fast Ethernet Line Card
pprSb96t	{ boards 168 }	*—Reserved—*
pprSb32fSC	{ boards 169 }	*—Reserved—*
pprSb32fMT	{ boards 170 }	AT-SB4352 32-Port (MT-RJ) Fast Ethernet Line Card
pprSb8fRJ	{ boards 172 }	AT-SB4411 8-Port (RJ-45) Gigabit Ethernet Line Card
pprSb8fSXSC	{ boards 173 }	AT-SB4451 8-port SX (SC) Gigabit Ethernet Line Card
pprSb8fSXMT	{ boards 174 }	AT-SB4452 8-port SX (MT-RJ) Gigabit Ethernet Line Card
pprSb8fLXSC	{ boards 175 }	AT-SB4461 8-Port LX (SC) Gigabit Ethernet Line Card
pprSb8fLXMT	{ boards 176 }	AT-SB4462 8-Port LX (MT-RJ) Gigabit Ethernet Line Card
pprAr410	{ boards 177 }	AT-AR410 (4 10/100 Ethernet switch ports, 1 10/100 Ethernet port, 1 RS-232 async port, 1 PIC expansion bay)
pprA40SC	{ boards 178 }	AT-A40/SC uplink module (1 100BASE-FX Multimode Fibre, SC connector)
pprA40MTRJ	{ boards 179 }	AT-A40/MT uplink module (1 100BASE-FX Multimode Fibre, MT-RJ connector)
pprA41SC	{ boards 180 }	AT-A41/SC uplink module (1 100BASE-FX Singlemode Fibre, SC connector)
pprA41MTRJ	{ boards 181 }	AT-A41/MT uplink module (1 100BASE-FX Singlemode Fibre, MT-RJ connector)
pprAr725	{ boards 182 }	AT-AR725 (2 10/100 LAN ports, 2 RS-232 async ports, 2 PIC expansion bays)
pprAr745	{ boards 183 }	AT-AR745 (2 10/100 LAN ports, 2 RS-232 async ports, 1 NSM expansion bay, 2 PIC expansion bays)
pprSb8GBIC	{ boards 184 }	AT-SB4441 8-slot GBIC line card
pprA42GBIC	{ boards 185 }	AT-A42/GBIC uplink module (1 1000BASE-X GBIC)
ppr9816GB	{ boards 186 }	AT-9816GB (16 1000BASE-X GBIC slots)
ppr9812T	{ boards 187 }	AT-9812T (12 10BASE-T/100BASE-TX/1000BASE-T ports, 4 1000BASE-X GBIC slots)
pprNsm048DS3	{ boards 188 }	AT-AR048 NSM (DS3 interface)
pprAr450	{ boards 191 }	CentreCOM AR450S (5 10/100 Ethernet switch ports, 2 10/100 Ethernet ports, 1 RS-232 async port)
pprAr450Dual	{ boards 192 }	AT-AR450S (5 10/100 Ethernet switch ports, 2 10/100 Ethernet ports, 2 RS-232 async ports)
pprSbExpander	{ boards 193 }	AT-SB4215 Bandwidth Expander
pprAr725DC	{ boards 194 }	AT-AR725 (DC power, 2 10/100 LAN ports, 2 RS-232 async ports, 2 PIC expansion bays)

Table B-4: Object identifiers for base CPU and expansion boards (continued)

Object	Object Identifier	Description
pprAr745DC	{ boards 195 }	AT-AR745 (DC power, 2 10/100 LAN ports, 2 RS-232 async ports, 1 NSM expansion bay, 2 PIC expansion bays)
pprAr410v2	{ boards 196 }	AT-AR410v2 (version 2, 4 10/100 Ethernet switch ports, 1 10/100 Ethernet port, 1 RS-232 async port, 1 PIC expansion bay)
pprAr410v3	{ boards 197 }	AT-AR410v3 (version 3, 4 10/100 Ethernet switch ports, 1 10/100 Ethernet port, 1 RS-232 async port, 1 PIC expansion bay)
pprIcmAr027	{ boards 198 }	AT-AR027 VoIP-FXS PIC
ppr8948EX	{ boards 202 }	AT-8948 (4 10/100 Ethernet switch ports, 4 1000BASE-X SFP uplink ports, 1 CompactFlash socket, 1 RS-232 async port)
ppr9816GBDC	{ boards 204 }	AT-9816GB (16 1000BASE-X GBIC slots, DC power)
ppr9812TDC	{ boards 205 }	AT-9812T (12 10BASE-T/100BASE-TX/1000BASE-T ports, 4 1000BASE-X GBIC slots, DC power)
pprIcmAr021v2s	{ boards 206 }	AT-AR021v2s BRI1 (S/T interface) PIC card
pprAtPwr01AC	{ boards 213 }	AT-PWR01 AC Power Supply Unit
pprAtPwr01DC	{ boards 214 }	AT-PWR01 DC Power Supply Unit
pprAtFan01	{ boards 215 }	AT-FAN01 Fan Only Module
ppr9924T	{ boards 218 }	AT-9924T Multi-layer IPv4 Gigabit Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 4 SFP Gigabit uplink ports
ppr9924SP	{ boards 219 }	AT-9924SP Multi-layer IPv4 Gigabit Switch, 24 SFP Gigabit uplink ports
ppr9924T4SP	{ boards 221 }	AT-9924T/4SP Multi-layer IPv4 and IPv6 Gigabit Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 4 SFP Gigabit uplink ports
pprAt8624T2M	{ boards 239 }	AT-8624T2M Layer 3 Fast Ethernet Switch

Table B-5: Object identifiers for interface types

Object	Object Identifier	Description
iftypes	{ objects 3 }	
ifaceEth	{ iftypes 1 }	Ethernet
ifaceSyn	{ iftypes 2 }	Synchronous
ifaceAsyn	{ iftypes 3 }	Asynchronous
ifaceBri	{ iftypes 4 }	BRI ISDN
ifacePri	{ iftypes 5 }	PRI ISDN
ifacePots	{ iftypes 6 }	POTS (voice)
ifaceGBIC	{ iftypes 7 }	GBIC (Gigabit Interface Converter)

Table B-6: Object identifiers for chip sets

Object	Object Identifier	Description
chips	{ objects 4 }	
chip68020Cpu	{ chips 1 }	68020 processor
chip68340Cpu	{ chips 2 }	68340 processor
chip68302Cpu	{ chips 3 }	68302 processor
chip68360Cpu	{ chips 4 }	68360 processor
chip860TCpu	{ chips 5 }	MPC 860T processor
chipRtc1	{ chips 21 }	Real Time Clock
chipRtc2	{ chips 22 }	Real Time Clock
chipRtc3	{ chips 23 }	Real Time Clock
chipRtc4	{ chips 24 }	Real Time Clock
chipRam1mb	{ chips 31 }	1 MB RAM
chipRam2mb	{ chips 32 }	2 MB RAM
chipRam3mb	{ chips 33 }	3 MB RAM
chipRam4mb	{ chips 34 }	4 MB RAM
chipRam6mb	{ chips 36 }	6 MB RAM
chipRam8mb	{ chips 38 }	8 MB RAM
chipRam12mb	{ chips 42 }	12 MB RAM
chipRam16mb	{ chips 46 }	16 MB RAM
chipRam20mb	{ chips 50 }	20 MB RAM
chipRam32mb	{ chips 62 }	32 MB RAM
chipFlash1mb	{ chips 71 }	1 MB FLASH memory
chipFlash2mb	{ chips 72 }	2 MB FLASH memory
chipFlash3mb	{ chips 73 }	3 MB FLASH memory
chipFlash4mb	{ chips 74 }	4 MB FLASH memory
chipFlash6mb	{ chips 76 }	6 MB FLASH memory
chipFlash8mb	{ chips 78 }	8 MB FLASH memory
chipPem	{ chips 120 }	Processor Enhancement Module

The sysinfo Group

The sysinfo Group has the object identifier prefix *sysinfo* ({ atRouter 3 }), and contains the following objects that describe generic system information:

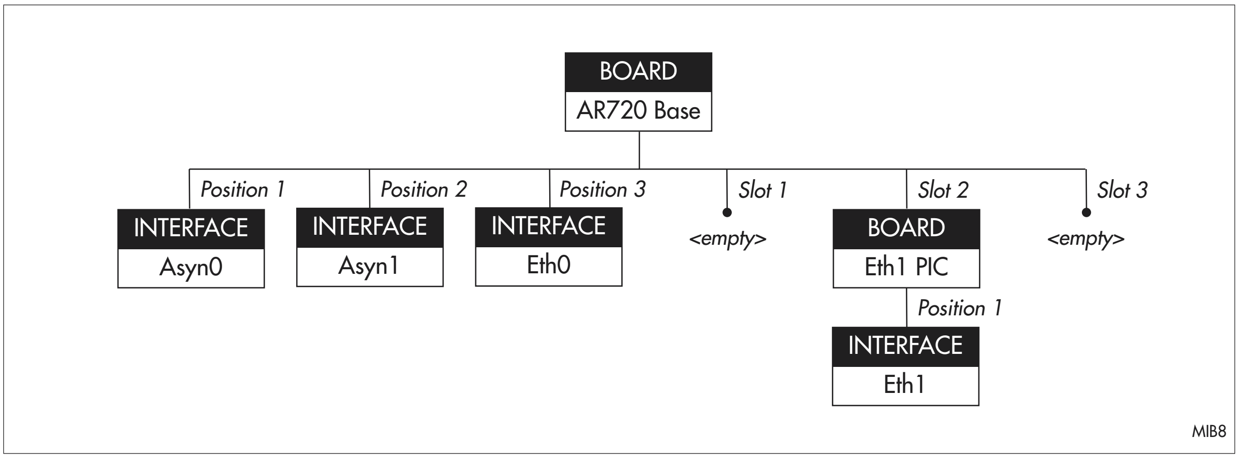
- *fanAndPs* ({ sysinfo 1 }) is a collection of objects and traps for managing the fan, power supply and temperature status of the router.
- *restart* ({ sysinfo 2 }) is used to perform a warm or cold restart of the router.
- *cpu* ({ sysinfo 3 }) is a collection of objects containing information about the CPU utilisation over different periods of time.
- *sysTemperature* ({ sysinfo 4 }) is a collection of objects containing information about temperature monitoring.
- *atContactDetails* ({ sysinfo 5 }) is the contact details for Allied Telesyn.

- *bbrNvs* ({ sysinfo 6 }) is information about the battery backed RAM non-volatile storage.

The arInterfaces Group

The arInterfaces Group contains objects that describe the boards, slots and physical interfaces in the router. A router consists of a number of “boards”. Each board may have a number of “positions”, each of which contains a single physical interface. Each board may also have a number of “slots”, which are places which can take other boards. Thus the physical construction of a router may be seen as a tree whose nodes are boards and interfaces, and whose links are positions and slots. Figure B-2 on page B-13 illustrates the object tree for an AR720 with two asynchronous ports and an Ethernet port on the base CPU board, and a single Ethernet PIC in Bay 1.

Figure B-2: The arInterfaces Group object tree for an AR720 with an Ethernet PIC in Expansion Bay 1



The function of the interface MIB tables is to represent this tree and to map elements in this tree to other MIB variables. Note that any given board has a fixed configuration of positions and slots; it is what is contained in the slots that give different hardware configurations.

The arInterfaces Group contains the following objects:

- *arBoardMaxIndex* ({ arInterfaces 1 }) is the maximum index of boards in *arBoardTable*. Boards have indices from 1 to the value of this object. There may be gaps in the sequence when the router has hot-swap capability. When the router has no hot-swap capability or no swapping has taken place since boot, the sequence of boards has no gaps. Index 1 is reserved for the main system board of the router.
- *arBoardTable* ({ arInterfaces 2 }) is a table of the boards in the router, indexed by board index (*arBoardIndex*). This table describes all of the physical boards present in the router. A board is defined as a separately removable circuit board with its own serial number.
- *arSlotTable* ({ arInterfaces 3 }) is a table of the slots in the router. This table is indexed by board index (*arSlotBoardIndex*) and slot index (*arSlotSlotIndex*) and gives the board index of the board occupying the given slot in the given board.
- *arInterfaceTable* ({ arInterfaces 4 }) is a table of the physical interfaces in the router. This table is indexed by board index (*arInterfaceBoardIndex*) and a board position (*arInterfacePosition*), and has an entry for each physical interface on the router.

- *arIfXTable* ({ arInterfaces 5 }) extends ifTable to allow interface variables not covered in the standard MIB, including interface statistics. This table is indexed by interface extension index (*arIfXIndex*), which maps directly to the equivalent *ifIndex* in *ifTable*, and has an entry for each physical interface on the router.

The Modules Group

The Modules Group contains objects that describe particular software modules in the router that are not covered by standard MIBs.

The Ethernet Group

The Ethernet Group has the object identifier prefix *ethernet* ({ modules 23 }), and contains the following objects that describe Ethernet interface(s) on the router:

- *ethIntTable* ({ ethernet 1 }) is a table of the Ethernet interfaces on the router. Each entry is a single Ethernet interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, and the duplex mode of the interface.

The ISDN Call Control Group

The ISDN Call Control Group contains objects that describe ISDN call definitions, active call details and call history on the router, and has the object identifier prefix *cc* ({ modules 37 }).

The following objects are defined:

- *ccDetailsTable* ({ cc 1 }) is a table of call details. Each entry contains the configuration for a single ISDN call.
- *ccCliListTable* ({ cc 2 }) is a table of all the CLI numbers from all CLI lists on the router, indexed by CLI list number. Each entry contains a single CLI number for matching against CLI information.
- *ccActiveCallTable* ({ cc 3 }) is a table of the active ISDN calls. Each entry contains the details for a single active call. Since active calls are created by internal router processes, this table is read-only.
- *ccCallLogTable* ({ cc 4 }) is a table of ISDN call log entries. Each entry contains log details of a single ISDN call. This table is read-only.
- *ccAttachmentTable* ({ cc 5 }) is a table of call detail attachment details, indexed by call detail index and list index of all the attachments from the user module. Each entry lists the index of the call details entry, the index of the attachment, the index of the active call for the attachment and the instance of the user module attached to the ISDN call. Since attachments are generated internally by router processes, this table is read-only.
- *ccBchannelTable* ({ cc 6 }) is a table of B channel attachment details, indexed by ISDN interface *ifIndex* and B channel index of all the attachments to ISDN B channels. Each entry lists the B channel *ifIndex*, the B channel number, whether the channel is allocated to a call, the type of call (if any), the index of the active call (if any), the priority of the call (if any), and the direction of the call (if any). Since attachments are generated internally by router processes, this table is read-only.

The BRI Group

The BRI Group contains objects that describe BRI interfaces on the router, and has the object identifier prefix *bri* ({ modules 41 }).

The following objects are defined:

- *briIntTable* ({ bri 1 }) is a table of the BRI interfaces on the router, indexed by *ifIndex*. Each entry is a single BRI interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, the operational mode of the interface (ISDN, TDM or mixed), a bit map of the channels in the interface dedicated to TDM and a bit map of the channels in the interface dedicated to ISDN.
- *briChanTable* ({ bri 2 }) is a table of the channels on BRI interfaces. Each entry is a single channel and lists the *ifIndex* of the interface, the channel index, the operational mode of the channel (ISDN, TDM or mixed), and the state (active or inactive) of the channel.

The PRI Group

The PRI Group contains objects that describe PRI interfaces on the router, and has the object identifier prefix *pri* ({ modules 42 }).

The following objects are defined:

- *priIntTable* ({ pri 1 }) is a table of the PRI interfaces on the router, indexed by *ifIndex*. Each entry is a single PRI interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, the operational mode of the interface (ISDN, TDM or mixed), a bit map of the channels in the interface dedicated to TDM, a bit map of the channels in the interface dedicated to ISDN, and the interface type (E1 or T1).
- *priChanTable* ({ pri 2 }) is a table of the channels on PRI interfaces. Each entry is a single channel and lists the *ifIndex* of the interface, the channel index, the operational mode of the channel (ISDN, TDM or mixed), and the state (active or inactive) of the channel.

The Loader Group

The Loader Group contains objects for managing the LOAD module which uses TFTP to download releases, patches, configuration scripts and other files from a TFTP server to NVS or flash storage in the router. Objects in this group have the object identifier prefix *loader* ({ modules 48 }).

The following objects are defined:

- *loadTable* ({ loader 1 }) is a table of load parameters. There are two entries, one for statically configured load information and one for dynamically configured load information. Static information is used when no dynamic information is available. Each entry lists the IP address of the TFTP server, the destination (NVS or flash), the name of the file to load, and the delay before loading.
- *loadStatus* ({ loader 2 }) is the status of the loader and is used to start and reset a load, or to report on the progress of a load.

The Install Group

The Install Group contains objects for managing the INSTALL module that controls the software release and patch running on the router. Objects in this group have the object identifier prefix *install* ({ modules 49 }).

The following objects are defined:

- *installTable* ({ install 1 }) is a table of install configurations and controls the software release and patch running on the router. Each entry contains a single install configuration and lists the install type (temporary, preferred or default), the device on which the release is stored (EPROM or flash), the release file name, the release version, the device on which the patch is stored (NVS or flash), the patch file name and the patch version.
- *installHistoryTable* ({ install 2 }) is a table of descriptions of events in the install history of the router. Each entry is a descriptive line that tells of part of the install history of the last router reboot.
- *configFile* ({ install 3 }) is the name of the file that the router configures from at boot. If the configuration file name is a zero length string, then no configuration file is defined in the router.
- *licenceTable* ({ install 4 }) is a table of release licences in the router. This table contains licences for releases of router software stored in flash. It lists the licence index, licence status, licenced software, version, licence password, and licence expiry date.
- *createConfigFile* ({ install 5 }) is the name of a file to create containing the current router configuration. A read from this variable returns the same as the variable *configFile*. To save the current configuration in the current configuration file, read *createConfigFile* first, then write the result back to *createConfigFile*. If this variable is written with the name of an existing file, the file is replaced with the current configuration.

The Trigger Group

The Trigger Group contains objects for managing triggers in the router. Objects in this group have the object identifier prefix *trigger* ({ modules 53 }).

The following objects are defined:

- *triggerLastTriggerActivated* ({ trigger 1 }) is the trigger number of the most recent trigger activated on the router.
- *triggerTrap* is a trap message generated when a trigger is activated.

The File Group

The File Group contains objects for managing the file system in the router. Objects in this group have the object identifier prefix *file* ({ modules 56 }).

The following objects are defined:

- *fileTable* ({ file 1 }) is a table of all the files in the router's non-volatile storage. Each entry contains the details of a single file and lists the file name, the device on which the file is stored (flash or NVS), the size of the file in bytes, the date and time the file was created, and the status of the file.
- *fileNumbers* ({ file 2 }) is the total number of files stored in flash and NVS on the router.

The Ping Group

The Ping Group contains objects for managing ping operations. Objects in this group have the object identifier prefix *ping* ({ modules 58 }).

The following objects are defined:

- *pingTable* ({ ping 1 }) is a table of static and dynamic ping information indexed by *pingIndex*, which takes the values *static(1)* or *dynamic(2)*. The static information is used when initiating a ping operation unless the dynamic information has been written with different values since the completion of the last ping.
- *pingStatus* ({ ping 2 }) is the ping status, used to start and stop a ping, or to report the status of the ping. *pingStatus* accepts and returns the values *start-running(1)* and *stop-stopped(2)*.
- *pingStatistics* ({ ping 3 }) is a collection of objects representing the results of a ping.
- *pingTrap* is a trap message generated when a ping has completed.

The DHCP Group

The DHCP Group contains objects for managing DHCP. Objects in this group have the object identifier prefix *dhcp* ({ modules 70 }).

The following objects are defined:

- *dhcpRangeTable* { dhcp 1 } is a table of DHCP ranges. Each entry in the table gives information about a single DHCP range currently configured on the router.
- *dhcpTrapVariable* { dhcp 2 } are special variables set up to act as reference points for variables sent in traps.

The Firewall Group

The Firewall Group contains objects describing traps generated by the firewall in the router. Objects in this group have the object identifier prefix *firewall* ({ modules 77 }).

The following objects are defined:

- *firewallTrapMessage* is the last TRAP message sent from the firewall. This variable is really just a placeholder for the object sent in the firewall TRAP, but can be read independently if required. Note however that a new firewall TRAP will cause this variable to be over written.
- *firewallTrap* is a trap message generated when the firewall detects an intrusion or attack. Firewall trap notifications are enabled using the command:

```
ENABLE FIREWALL NOTIFY=SNMP
```

The Load Balancer Group

The Load Balancer Group contains objects describing the current configuration and status of load balancing operations on the router. Objects in this group have the object identifier prefix *lb* ({ modules 104 }).

The following objects are defined:

- *lbShowGlobalTable* { lb 1 } is a table of general configuration and status information of all of the virtual balancers configured on the device.
- *lbShowResTable* { lb 2 } is a table of general information about a given resource.
- *lbShowResPoolTable* { lb 3 } is a table of general information about a given resource pool.
- *lbShowVirtBalTable* { lb 4 } is a table of general information about a given virtual balancer.
- *lbShowAffTable* { lb 5 } is a list of affinity tables for all configured virtual balancers on the device.
- *lbShowConTable* { lb 6 } is a table listing information about all of the TCP connections currently open to a given virtual balancer.

The OSI Group

The OSI Group contains objects for managing the OSI CLNS and ISIS protocols. Objects in this group have the object identifier prefix *osi* ({ modules 94 }).

The following objects are defined:

- *isisSystem* ({ osi 1 }) is a collection of tables containing general system objects:
  - The System Table *isisSysTable* ({ isisSystem 1 } is a table of the instances of the Integrated IS-IS protocol on the system.
  - The Level 1 Manual Area Address Table *isisManAreaAddrTable* ({ isisSystem 2 }) is a table of area addresses manually configured for each instance of the Integrated IS-IS protocol.
  - The Level 1 Area Address Table *isisAreaAddrTable* ({ isisSystem 3 }) is a table of the area addresses reported in all Level 1 LSPs received by this Intermediate System.
  - The System Protocol Supported Table *isisSysProtSuppTable* ({ isisSystem 4 }) is a table of the manually configured set of protocols supported by each instance of the Integrated ISIS protocol.
  - The Level 2 Summary Address Table *isisL2SummAddrTable* ({ isisSystem 5 }) is a table of summary addresses manually configured for each Level 2 instance of IP Integrated ISIS on the system.
- *isisCirc* ({ osi 2 }) is a collection of tables of objects for managing circuits:
  - The Circuit Table *isisCircTable* ({ isisCirc 1 }) is a table of the broadcast or point-to-point interfaces on the system.
  - The Circuit IS Table *isisCircISTable* ({ isisCirc 2 }) is a table of objects controlling the operation of the IS functions of the ES-IS protocol (ISO 9542) on each circuit over which ISIS is run.
  - The Circuit CLNS Table *isisCircClnsTable* ({ isisCirc 2 }) is a table of objects controlling the operation of the IS functions of the CLNS protocol (ISO 8473) on each circuit over which ISIS is run.

- *sisISAdj* ({ osi 3 }) is a collection of objects for managing adjacencies to routers maintained by the Integrated IS-IS protocol:
  - The IS Adjacency Table *isisISAdjTable* ({ isisISAdj 1 }) is a table of objects for each adjacency to an IS.
  - The IS Adjacency Area Address Table *isisISAdjAreaAddrTable* ({ isisISAdj 2 }) is a table of objects for area addresses of neighbouring Intermediate Systems as reported in IIH PDUs.
  - The IS Adjacency IP Address Table *isisISAdjIPAddrTable* ({ isisISAdj 3 }) is a table of objects for IP addresses of neighboring Intermediate Systems as reported in received IIH PDUs.
  - The IS Adjacency Protocol Supported Table *isisISAdjProtSuppTable* ({ isisISAdj 4 }) is a table of objects for the protocols supported by neighboring Intermediate Systems as reported in received IIH PDUs.
- *isisESAdj* ({ osi 4 }) is a collection of objects for managing the ES-IS protocol:
  - The ES Adjacency Table *isisESAdjTable* ({ isisESAdj 1 }) is a table of neighboring End Systems as reported in ESH PDUs or as configured manually.
- *isisReachAddr* ({ osi 5 }) is a collection of objects for managing reachable addresses:
  - The Reachable Address Table *isisRATable* ({ isisReachAddr 1 }) is a table of objects for reachable addresses (NSAP or address prefix) manually configured on the system or learned through another protocol.
- *isisCLNPDest* ({ osi 7 }) is a collection of objects for managing CLNP destinations:
  - The Level 1 CLNP Destination Table *isisL1CLNPDestTable* ({ isisCLNPDest 1 }) records information about each end system ID destination known to the Intermediate System in the system's level 1 forwarding database.
  - The Level 2 CLNP Destination Table *isisL2CLNPDestTable* ({ isisCLNPDest 2 }) records information about each NSAP Address Prefix known to the Intermediate System.
  - The Level 3 CLNP Destination Table *isisL3CLNPDestTable* ({ isisCLNPDest 3 }) records information about each Reachable Address Prefix known to the Intermediate System.
- *osiVirtualInstance* { osi 8 }) defines the OSI Virtual Router Instance which will respond to SNMP commands. This defaults to instance 0. If an instance is not enabled then it cannot respond to SNMP commands.

The Port Authentication Group

The Port Authentication group contains additional objects for managing systems using a multi-supplicant configuration that are not included in the Port Access Control (IEEE 802.1X) MIB ("Port Access Control MIB" on page B-35).

This MIB is implemented on the AR410 only.

Objects in this group have the object identifier prefix *atrPaeMib* { modules(4) portAuth(118) }, and are identical to the objects in IEEE Standard 8021paeMIB ({ ieee802dot1mibs 1 }, "Port Access Control MIB" on page B-35), except that:

- The names of the objects are all prefixed with "*atr*".
- There is an additional object to represent the MAC address of a supplicant.

The following objects are defined:

- *atrDot1xPaePortTable* is a table of system level information (port number, protocol version, and initialisation control) for each port supported by the Port Access Entity.
- *atrDot1xAuthConfigTable* is a table of configuration objects for the Authenticator PAE associated with each port. An entry appears in this table for each port that may authenticate access to itself.
- *atrDot1xAuthStatsTable* is a table of statistics objects for the Authenticator PAE associated with each Port.

All tables in this MIB are indexed by port number (*atrDot1xPaePortNumber*) and supplicant MAC address (*atrDot1xPaePortSuppMacAddress*).

This MIB maintains Authenticator PAE settings for each supplicant attached to an Authenticator PAE. The default values for the port as a whole are stored in the standard Port Access Control MIB. Whenever a new Allied Telesyn specific Port Authentication MIB instance is created due to the presence of a new supplicant attached to a port, its initial values will be copied from the standard Port Access Control MIB entry for the applicable port. The enterprise MIB variables may then be overridden by the user on a per-supplicant basis.

The Stack Group

The stack group contains objects for managing stack members. Objects in this group have the object identifier stack ({ modules 120 }).

The following objects are defined:

- *stackId* ({ stack 1 })

  The stack ID variable identifies the given device as belonging to the stack with the given stack ID. The default stack ID is 1. To change the stack ID, stacking must be disabled. Once the stack ID is changed and stacking enabled again, the member joins an existing stack with the same stack ID or forms a new stack with its stack ID.

- *stackSnmpHost* ({ stack 2 })

  The SNMP host variable specifies the host ID of the stack member that should service SNMP requests. It defaults to the highest host ID that is active in the stack, and is re-evaluated whenever the stack membership changes. This ensures that the variable always has a value that corresponds to a valid stack member.

- *stackStatus* ({ stack 3 })

  The stack status variable specifies the current status of the stack module on the current SNMP host. The default is disabled (1).

- *stackInterface* ({ stack 4 }) The stack interface variable is used to specify the interface on the current SNMP host that should be used to send and receive stack messages. The default is none and can be replaced by an existing VLAN.

- *stackAuth* ({ stack 5 })

  The stack authentication variable specifies whether stacking messages should be authenticated with the password given as the value of the stack password variable. By default, the authentication method is none (0). If plaintext (1) is specified, a cleartext password authenticates stacking messages. If MD5 (2) is specified, an encrypted password authenticates stacking protocol packets. The authentication type can be modified only when stacking is disabled on the device.

- *stackPassword* ({ stack 6 })

  The stack password variable holds a text string that authenticates stack messages. By default, this variable holds an empty string. This must be changed before setting the stack authentication method to plaintext or MD5.

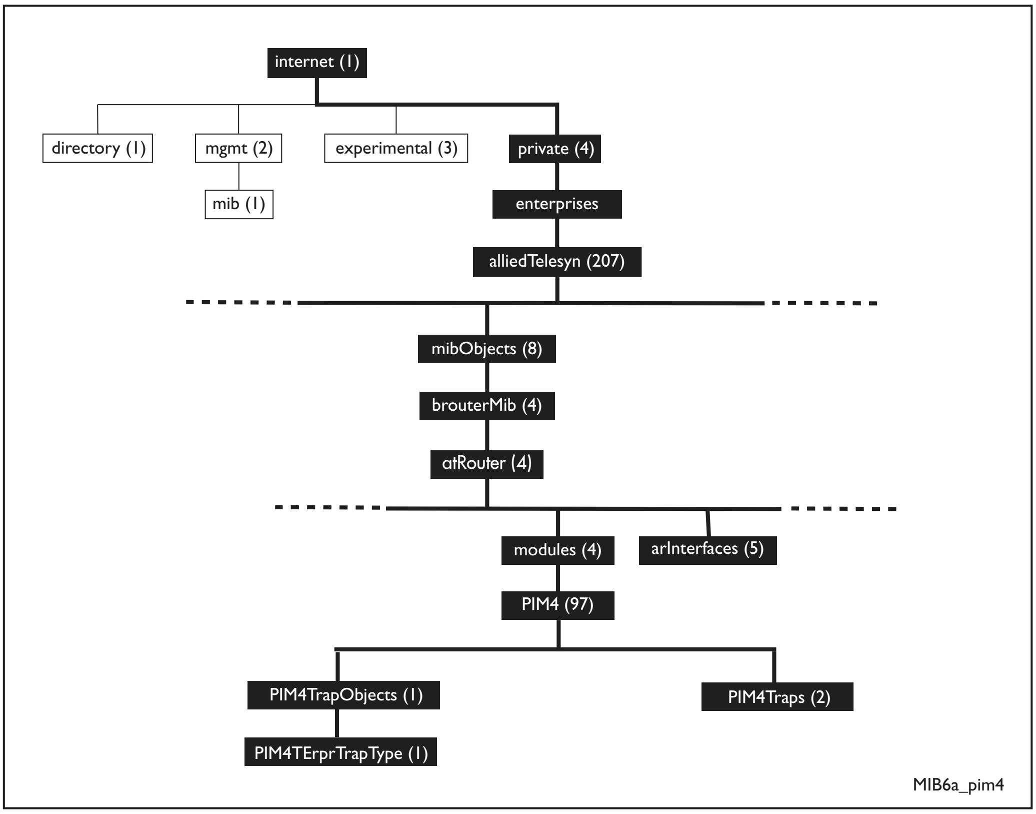
- *stackStats* ({ stack 7 })

  Stack statistics is a collection of objects representing the occurrences of specific events to a specific stack host.

- *stackMemberTable* ({ stack 8 })

  This table displays details about each of the stack members that are present in the stack to which the current SNMP host belongs. The details provided are the host ID, MAC address, designated and backup master, and state of each host.

The PIM4 Group

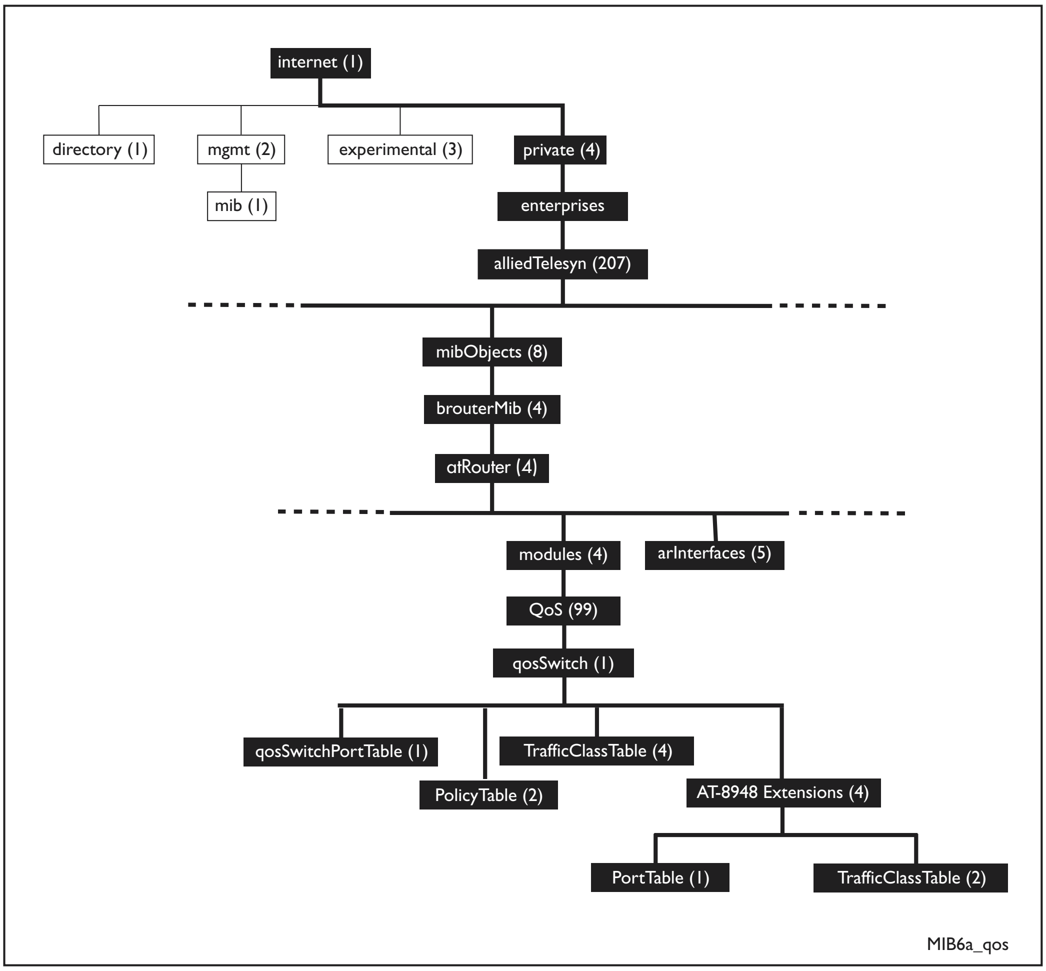


The PIM4 group contains a collection of objects for managing PIM on IPv4. Objects in this group have the object identifier *pim4* ({ modules 97 }).

The following objects are defined.

- *pim4TrapObjects* ({ pim4 1 }) contains a single object:
  - *pim4ErrorTrapType* is the type of the last error that resulted in an error trap being sent.
- *pim4Traps* ({ pim4 2 }) is a collection of traps for the following PIM events:
  - *pim4NeighbourAddedTrap* ({ pim4Traps 1 }) is generated when a PIM neighbour has been added.
  - *pim4NeighbourDeletedTrap* ({ pim4Traps 2 }) is generated when a PIM neighbour has been deleted.
  - *pim4InterfaceUpTrap* ({ pim4Traps 3 }) is generated when a PIM interface has been enabled and is active.
  - *pim4InterfaceDownTrap* ({ pim4Traps 4 }) is generated when a PIM interface has been disabled and is inactive.
  - *pim4ErrorTrap* ({ pim4Traps 5 }) is generated for a PIM error. The error is defined by pim4ErrorTrapType.

The QoS Group



The QoS group contains a collection of objects for managing QoS.

The following objects are defined:

- *qosSwitch* ({ qos 1 }) is a collection of generic objects for managing QoS:
    - *qosSwitchPortTable* ({ qosSwitch 1)} is a table of generic port configurations for QoS, indexed by port index.
    - *qosSwitchPolicyTable* ({ qosSwitch 2)} is a table of QoS policy configurations, indexed by policy index.
    - *qosSwitchTrafficClassTable* ({ qosSwitch 3)} is a table of traffic class configurations, indexed by traffic class index.
- *qosSwitch8948* ({ qosSwitch 4 }) is a collection of objects for managing QoS features specific to the AT-8948:
    - *qosSwitch8948PortTable* ({ qosSwitch8948 1 }) is a table of port configurations for QoS, indexed by port index.
    - *qosSwitch8948TrafficClassCountersTable* ({ qosSwitch8948 2 }) is a table of traffic class counters indexed by port, policy and traffic class.

The Software QoS Group

The software QoS group contains objects for managing software QoS. Objects in this group have the object identifier *sqos* ({ modules 123 }).

The following objects are defined:

- *sqosModuleTable* ({ sqos 1 }) is a table that displays the module ID and whether the module is enabled or disabled.
- *sqosPolicyTable* ({ sqos 2 }) is a table of information about each policy that is configured on the router.
- *sqosTrafficTable* ({ sqos 3 }) is a table of information about each traffic class that is configured on the router, including a list of the DAR objects that are attached to each traffic class.
- *sqosMeterTable* ({ sqos 4 }) is a table of information about each meter that is configured on the router. Meters measure bandwidth compliance and assign packets to bandwidth classes depending on their compliance.
- *sqosRedTable* ({ sqos 5 }) is a table of information about each RED curve set that is configured on the router, including the default RED curve sets.
- *sqosDscpMapTable* ({ sqos 6 }) is a table that displays the ID number and description of each DSCP map that is configured on the router. Software QoS uses DSCP maps when
    - premarking, which occurs at the start of the QoS processing. Premarking assigns a packet to a bandwidth class and/or gives it a new DSCP, on the basis of its current DSCP.
    - remarking, which occurs after bandwidth metering. Remarking gives a packet a new DSCP and/or assigns it to a bandwidth class, on the basis of its DSCP and meter bandwidth class.
- *sqosDscpMapRowTable* ({ sqos 7 }) is a table of information about each DSCP map that is configured on the router. Note that it only displays the lines which are different from the default—lines in which bandwidth class and/or DSCP changes.
- *sqosDarTable* ({ sqos 8 }) is a table of information about each Dynamic Application Recognition (DAR) object that is configured on the router. DAR objects recognise sessions such as VoIP phone calls and real-time video and create dynamic classifiers to match traffic belonging to these sessions.
- *sqosInterfaceTable* ({ sqos 9 }) is a table that lists the policies and DAR objects attached to interfaces on the router.
- *sqosPolicyCounterTable* ({ sqos 10 }) is a table that displays policy counters as packets and bytes, and the interface and traffic direction the policy applies to.
- *sqosTcCounterTable* ({ sqos 11 }) is a table that displays traffic class counters as packets and bytes, and the interface and traffic direction the traffic class applies to. It includes the queue length, the number of packets and bytes assigned to each bandwidth class, and the number dropped by RED.
- *sqosClfrCounterTable* ({ sqos 12 }) is a table that displays classifier counters as packets and bytes, and the interface and traffic direction the classifier applies to.
- *sqosDarCounterTable* ({ sqos 13 }) is a table that displays DAR object counters, including the number of setup messages, current sessions, and total sessions recognised by each DAR object.

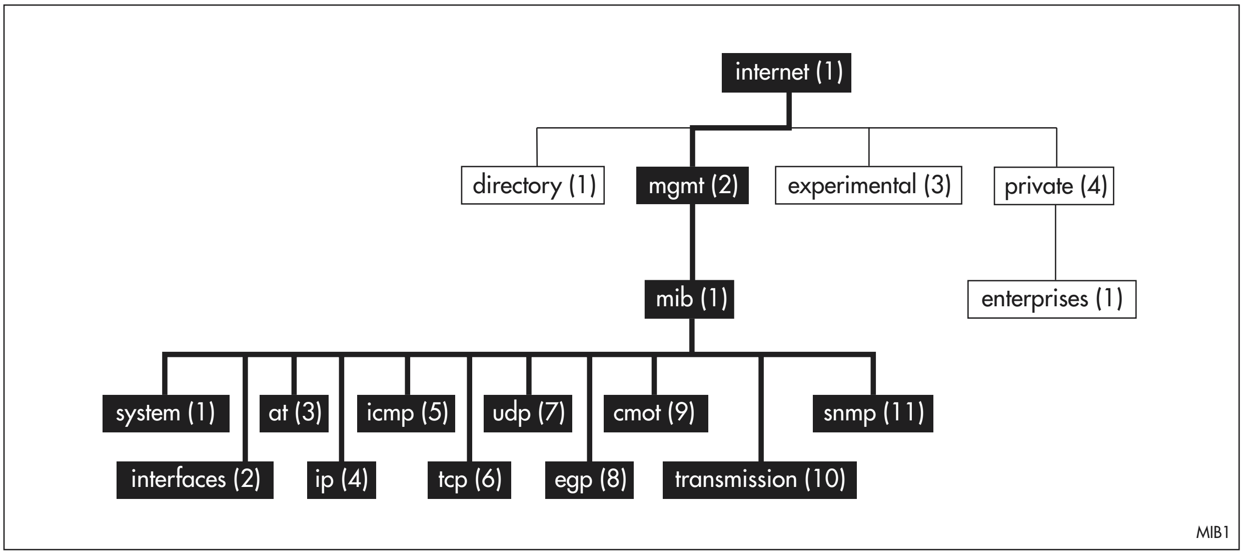
- *sqosTraps* ({ sqos 100 }) is a collection of traps for the following software QoS events:
    - *sqosTrapPaused* ({ sqosTraps 1 }) is generated when a traffic class temporarily stops processing packets because the bandwidth allocation is exceeded.
    - *sqosTrapUnpaused* ({ sqosTraps 2 }) is generated when a paused traffic class starts processing packets again.
    - *sqosTrapQLimitExceeded* ({ sqosTraps 3 }) is generated when a queue overflows.
    - *sqosTrapDCLimitExceeded* ({ sqosTraps 4 }) is generated when the allowed number of dynamic classifiers is exceeded.

MIB-II MIB

RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II,* defines the core set of managed objects for TCP/IP-based internets, and supersedes MIB-I defined in RFC 1156.

Objects defined in MIB-II reside in the mib(1) sub-tree (Figure B-3 on page B-26) and have the object identifier prefix *mib-2* ({ mgmt 1 }).

Figure B-3: The MIB-II sub-tree of the Internet-standard Management Information Base (MIB)



The objects in MIB-II are arranged into 11 groups:

- The *System* group contains objects that describe contact, administrative, location, and service information for the entity.
- The *Interfaces* group contains objects that describe the interfaces on the entity. Each interface is thought of as being attached to a `subnetwork'. Note that this term should not be confused with *subnet*, which refers to an addressing partitioning scheme used in the Internet suite of protocols.
- The *Address Translation* group contains objects that describe the translation between network addresses (e.g. IP addresses) and subnetwork-specific, or physical, addresses on the entity.
- The *IP* group contains objects that describe the entity’s IP addressing scheme, IP routing table, IP address translation and counters for the IP forwarding process.
- The *ICMP* group contains objects that record traffic statistics for the ICMP protocol on the entity.
- The *TCP* group contains objects that describe the TCP protocol, active TCP connections and TCP traffic counters on the entity.
- The *UDP* group contains objects that describe the UDP protocol, active UDP connections and UDP traffic counters on the entity.
- The *EGP* group contains objects that describe the entity’s EGP neighbours and traffic counters for the EGP protocol on the entity.
- The *CMOT* group is preserved for historical reasons and is not used for management of the entity.
- The *Transmission* group contains objects that describe the different types of transmission media supported on the entity.

- The *SNMP* group contains objects that describe the SNMP protocol on the entity and traffic counters for the SNMP protocol on the entity.

These groups are the basic unit of conformance: if any of the objects in a group are applicable to an implementation, then it must implement all objects in that group.

Implementation

All router models implement all groups in MIB-II. However, the implementation of some objects differs from RFC 1213 (Table B-7 on page B-27). In particular, some read-write objects are implemented as read-only.

Table B-7: MIB-II implementation variations

Object Name	Object ID	Access	Implementation
ifAdminStatus	{ 1.3.6.1.2.1.2.2.1.7 }	Read-write	Read-write for PPP, VLAN & Syn interfaces. Read-only for all other interfaces.
atIfIndex	{ 1.3.6.1.2.1.3.1.1.1 }	Read-write	Read-only
atPhysAddress	{ 1.3.6.1.2.1.3.1.1.2 }	Read-write	Read-only
atNetAddress	{ 1.3.6.1.2.1.3.1.1.3 }	Read-write	Read-only
ipRouteIfIndex	{ 1.3.6.1.2.1.4.21.1.2 }	Read-write	Read-only
ipRouteMetric1	{ 1.3.6.1.2.1.4.21.1.3 }	Read-write	Read-only
ipRouteMetric2	{ 1.3.6.1.2.1.4.21.1.4 }	Read-write	Read-only
ipRouteMetric3	{ 1.3.6.1.2.1.4.21.1.5 }	Read-write	Read-only
ipRouteMetric4	{ 1.3.6.1.2.1.4.21.1.6 }	Read-write	Read-only
ipRouteNextHop	{ 1.3.6.1.2.1.4.21.1.7 }	Read-write	Read-only
ipRouteType	{ 1.3.6.1.2.1.4.21.1.8 }	Read-write	Read-only
ipRouteAge	{ 1.3.6.1.2.1.4.21.1.10 }	Read-write	Read-only
ipRouteMask	{ 1.3.6.1.2.1.4.21.1.11 }	Read-write	Read-only
ipRouteMetric5	{ 1.3.6.1.2.1.4.21.1.12 }	Read-write	Read-only
ipNetToMediaIfIndex	{ 1.3.6.1.2.1.4.22.1.1 }	Read-write	Read-only
ipNetToMediaPhysAddress	{ 1.3.6.1.2.1.4.22.1.2 }	Read-write	Read-only
ipNetToMediaNetAddress	{ 1.3.6.1.2.1.4.22.1.3 }	Read-write	Read-only
ipNetToMediaType	{ 1.3.6.1.2.1.4.22.1.4 }	Read-write	Read-only
tcpConnState	{ 1.3.6.1.2.1.6.13.1.1 }	Read-write	Read-only
egpNeighEventTrigger	{ 1.3.6.1.2.1.8.5.1.15 }	Read-write	Read-only
snmpEnableAuthenTraps	{ 1.3.6.1.2.1.11.30 }	Read-write	Read-only
ipRouteDest	{ 1.3.6.1.2.1.4.21.1.1 }	Read-write	Read-only

IP Forwarding Table MIB

RFC 2096, *IP Forwarding Table MIB,* defines a portion of the Management Information Base (MIB) for managing Classless Inter-Domain Routing (CIDR).

Objects defined in this MIB reside in the mib(1) sub-tree, under the IP Group defined in MIB-II (Figure B-3 on page B-26) and have the object identifier prefix *ipForward* ({ ip 24 }).

The IP Forwarding Table MIB consists of the following objects:

- The *ipForwardNumber* object contains the number of current *ipForwardTable* entries that are not invalid.
- The IP Forwarding Table (*ipForwardTable*) contains objects that obsolete and replace objects in the ipRoute Table in MIB-I and MIB-II, and additional objects that describe the autonomous system of the next hop, multiple next hop support, and policy routing support.
- The *ipCidrRouteNumber* object contains the number of current ipCidrRouteTable entries that are not invalid.
- The IP CIDR Route Table (*ipCidrRouteTable*) contains objects that obsolete and replace objects in the ipRoute Table in MIB-I and MIB-II and the IP Forwarding Table, and additional objects that describe the autonomous system of the next hop, multiple next hops, policy routing, and Classless Inter-Domain Routing.

Implementation

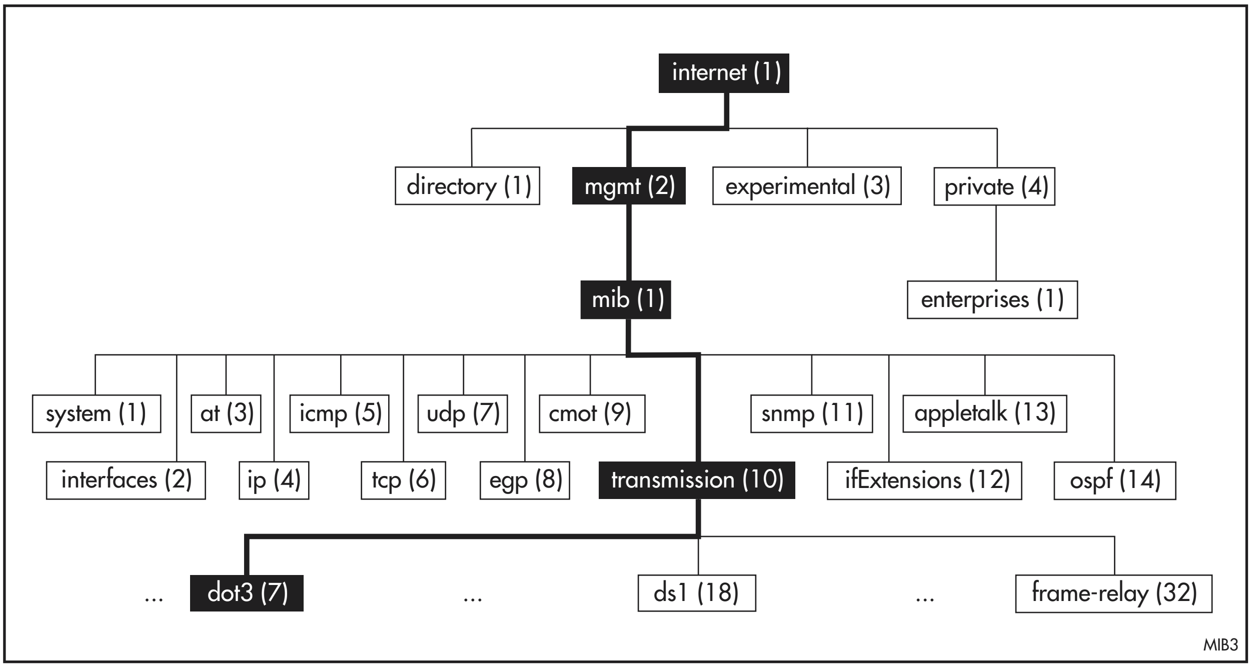
All router models implement all objects in the Ethernet-like Interface Types MIB.

Ethernet-like Interface Types MIB

RFC 1643, *Definitions of Managed Objects for the Ethernet-like Interface Types*, defines a portion of the Management Information Base (MIB) for managing Ethernet-like objects. The detail for Ethernet specified in RFC 1643 overrides generic values given in RFC 1573.

Objects defined in this MIB reside in the mib(1) sub-tree, under the Transmission Group defined in MIB-II (Figure B-4 on page B-29) and have the object identifier prefix *dot3* ({ transmission 7 }).

Figure B-4: The Ethernet-like interface types sub-tree of the Internet-standard Management Information Base (MIB)



Instances of these object types represent attributes of Ethernet-like interfaces. At present, Ethernet-like media are identified by three values of the *ifType* object in the Internet Standard MIB:

- ethernet-csmacd(6)
- iso88023-csmacd(7)
- starLan(11)

For these interfaces, the value of the *ifSpecific* variable in MIB-II has the object identifier value *dot3* ({ transmission 7 }).

The objects in this MIB are organised into two groups:

- The *Ethernet-like Statistics* group contains objects that record statistics for Ethernet-like interfaces on the entity.
- The *Ethernet-like Collision Statistics* group contains objects that describe collision histograms for Ethernet-like interfaces on the entity.

Implementation

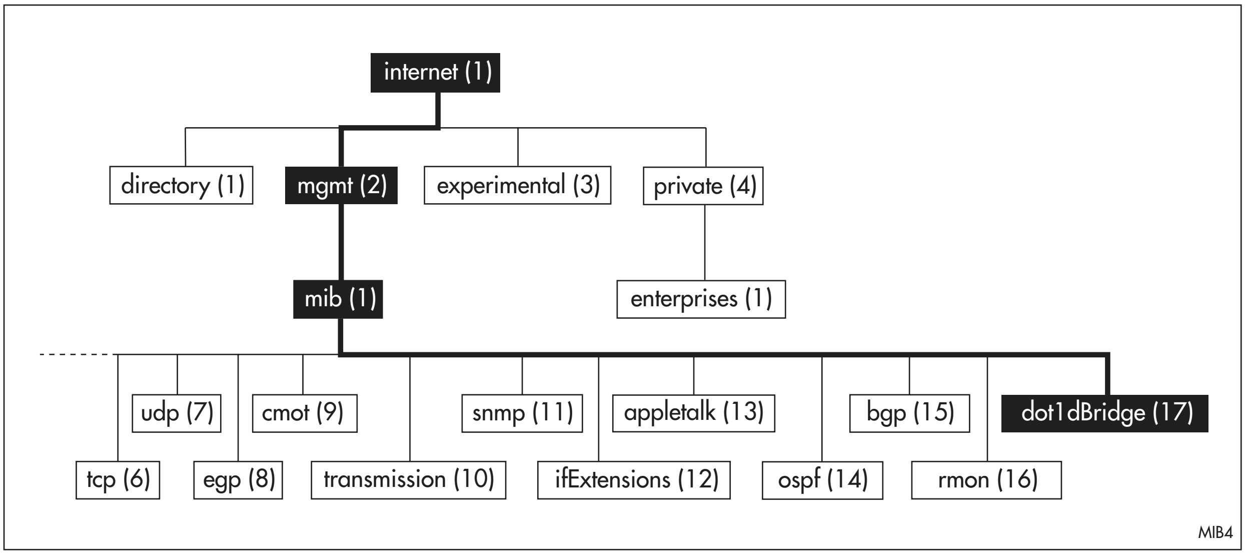
All router models implement all objects in the Ethernet-like Interface Types MIB.

Bridge MIB

RFC 1493, *Definitions of Managed Objects for Bridges,* defines a portion of the Management Information Base (MIB) for managing MAC bridges based on the IEEE Standard 802.1D-1990 standard between Local Area Network (LAN) segments. Provisions are made for support of transparent bridging and for the connection of bridges by subnetworks other than LAN segments.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-5 on page B-30) and have the object identifier prefix *dot1dBridge* ({ mib-2 17 }).

Figure B-5: The bridge sub-tree of the Internet-standard Management Information Base (MIB)



There are two major modes defined for this bridging; transparent and source route. The transparent method of bridging is defined in the draft IEEE Standard 802.1d specification (*ISO DIS 10038 MAC Bridges*). This MIB defines those objects needed for the management of a bridging entity operating in the transparent mode, as well as some objects applicable to all types of bridges.

The MIB is composed of the following sections:

- The *dot1dBase* Group is mandatory and contains the objects which are applicable to all types of bridges.
- The *dot1dStp* Group contains objects that denote the bridge's state with respect to the Spanning Tree Protocol. If a node does not implement the Spanning Tree Protocol, this group is not implemented.
- The *dot1dSr* Group contains objects that describe the entity's state with respect to source route bridging. If source routing is not supported this group is not implemented. This group is applicable to source route only and SRT bridges.
- The *dot1dTp* Group contains objects that describe the entity's state with respect to transparent bridging. If transparent bridging is not supported this group is not implemented. This group is applicable to transparent only and SRT bridges.
- The *dot1dStatic* Group contains objects that describe the entity's state with respect to destination-address filtering. If destination-address filtering is not supported this group is not implemented. This group is applicable to any type of bridge that performs destination-address filtering.

It is assumed that a bridge implementing this MIB also implements (at least) the *system* group and the *interfaces* group defined in MIB-II. The *system* group contains objects that apply to the entity as a whole irrespective of whether the entity's sole functionality is bridging, or whether bridging is a subset of the entity's functionality. The *interfaces* group contains information on an entity's interfaces, where each interface is thought of as being attached to a subnetwork. The term segment is used in this MIB to refer to such a subnetwork, whether it be an Ethernet segment, a ring, a WAN link, or an X.25 circuit.

Implicit in this Bridge MIB is the notion of ports on a bridge. Each of these ports is associated with one interface of the *interfaces* group, and in most situations, each port is associated with a different interface. However, there are situations in which multiple ports are associated with the same interface. An example of such a situation would be several ports each corresponding one-to-one with several X.25 virtual circuits but all on the same interface. Each port is uniquely identified by a port number. A port number has no mandatory relationship to an interface number, but in the simple case a port number has the same value as the corresponding interface's interface number. Port numbers are in the range (1..*dot1dBaseNumPorts*).

Some entities perform other functionality (such as routing) as well as bridging through the sending and receiving of data on their interfaces. In such situations, a subset of the data sent/received on an interface is within the domain of the entity's bridging functionality. The Bridge MIB and its counters are applicable only to that subset of the data on an entity's interfaces that is sent/received for a protocol being bridged. All such data is sent/received via the ports of the bridge.

Implementation

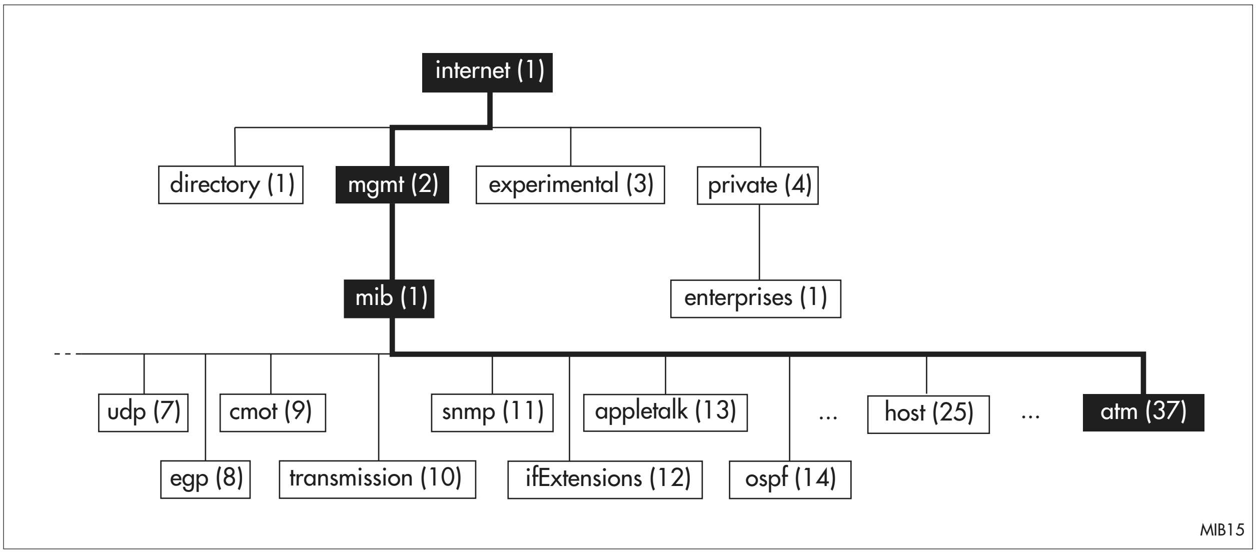
All router models implement all groups in the Bridge MIB.

ATM MIB

RFC 2515, *Definitions of Managed Objects for ATM Management*, defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes objects used for managing ATM-based interfaces, devices, networks and services.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-6 on page B-32) and have the object identifier prefix atmMib ({ mib-2 37 }).

Figure B-6: The atm sub-tree of the Internet-standard Management Information Base (MIB)



Implementation

The router complies with this MIB according to compliance statement *atmMIBCompliance2*. The following ATM MIB groups are implemented on routers with ATM support, with variations as shown in Table B-8:

- The *atmInterfaceConfGroup2* group (mandatory)
- The *atmTrafficDescrGroup2* group (mandatory)
- The *atmVccTerminationGroup2* group
- The *aal5VccGroup* group.

Table B-8: ATM MIB implementation variations

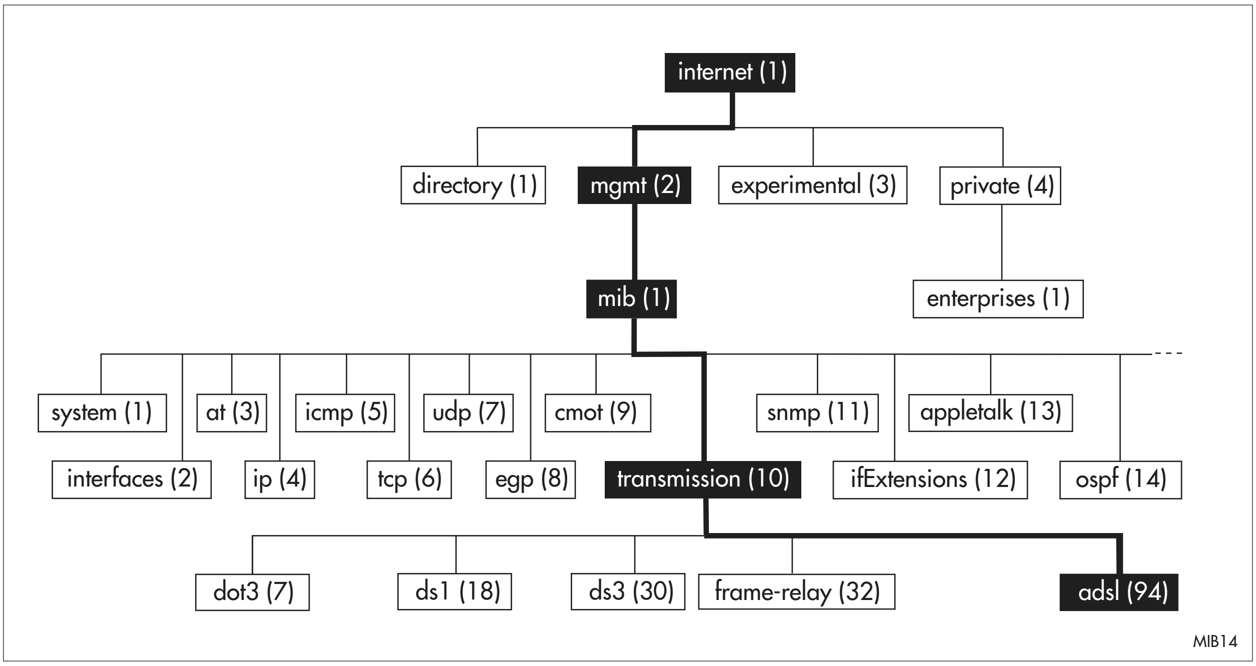
Object Name	Object ID	Access	Implementation
atmInterfaceMaxVpcs	{1.3.6.1.2.1.37.1.2.1.1}	Read-write	Read-only
atmInterfaceMaxVccs	{1.3.6.1.2.1.37.1.2.1.2}	Read-write	Read-only
atmInterfaceMaxActiveVpiBits	{1.3.6.1.2.1.37.1.2.1.5}	Read-write	Read-only
atmInterfaceMaxActiveVciBits	{1.3.6.1.2.1.37.1.2.1.6}	Read-write	Read-only
atmInterfaceIlmiVpi	{1.3.6.1.2.1.37.1.2.1.7}	Read-write	Read-only
atmInterfaceIlmiVci	{1.3.6.1.2.1.37.1.2.1.8}	Read-write	Read-only
atmInterfaceMyNeighborIpAddress	{1.3.6.1.2.1.37.1.2.1.11}	Read-write	Read-only
atmInterfaceMyNeighborIfName	{1.3.6.1.2.1.37.1.2.1.12}	Read-write	Read-only
atmInterfaceSubscrAddress	{1.3.6.1.2.1.37.1.2.1.15}	Read-write	Read-only
aal5VccSarTimeOuts	{1.3.6.1.2.1.37.1.12.1.4}	—	Not implemented

ADSL Line MIB

RFC 2662, *Definitions of Managed Objects for ADSL Lines*, defines a standard SNMP MIB for ADSL lines based on the ADSL Forum standard data model. The ADSL standard describes ATU-C and ATU-R as two sides of the ADSL line. The MIB covers both ATU-C and ATU-R agent's perspectives. Each instance defined in the MIB represents a single ADSL line. The router is an ATU-R, as it is located at the customer premises end of the line.

The objects defined in this MIB reside in the mib(1) sub-tree (Figure B-6 on page B-32), under the Transmission Group defined in MIB-II and are part of the ADSL MIB { transmission 94 }. The MIB object identifier is adslLineMib { adslMIB 1 }.

Figure B-7: The adsl sub-tree of the Internet-standard Management Information Base (MIB)



Implementation

The router complies with this MIB according to compliance statement *adslLineMibAturCompliance*. All routers with ADSL ports support the following ADSL Line MIB groups for ATU-Rs:

- All objects in *adslAturLineGroup*.
- In *adslAturPhysicalGroup*, the objects:
  - *adslAturCurrSnrMgn*
  - *adslAturCurrAtn*
  - *adslAturCurrStatus*
  - *adslAtucCurrStatus*
- All objects in *AdslAturAtucPhysPerfRawCounterGroup* except *adslAtucPerfInits*.
- All objects in *adslAturAtucPhysPerfIntervalGroup* except
  - *adslAtucPerfCurr15MinInits*

  - *adslAtucPerfCurr1DayInits*
  - *adslAtucPerfPrev1DayInits*
  - *adslAtucIntervalInits*
- All objects in *adslAturAturPhysPerfRawCounterGroup* except *adslAturPerfLprs*.
- All objects in *adslAturAturPhysPerfIntervalGroup* except
  - *adslAturPerfCurr15MinLprs*
  - *adslAturPerfCurr1DayLprs*
  - *adslAturPerfPrev1DayLprs*
  - *adslAturIntervalLprs*

Port Access Control MIB

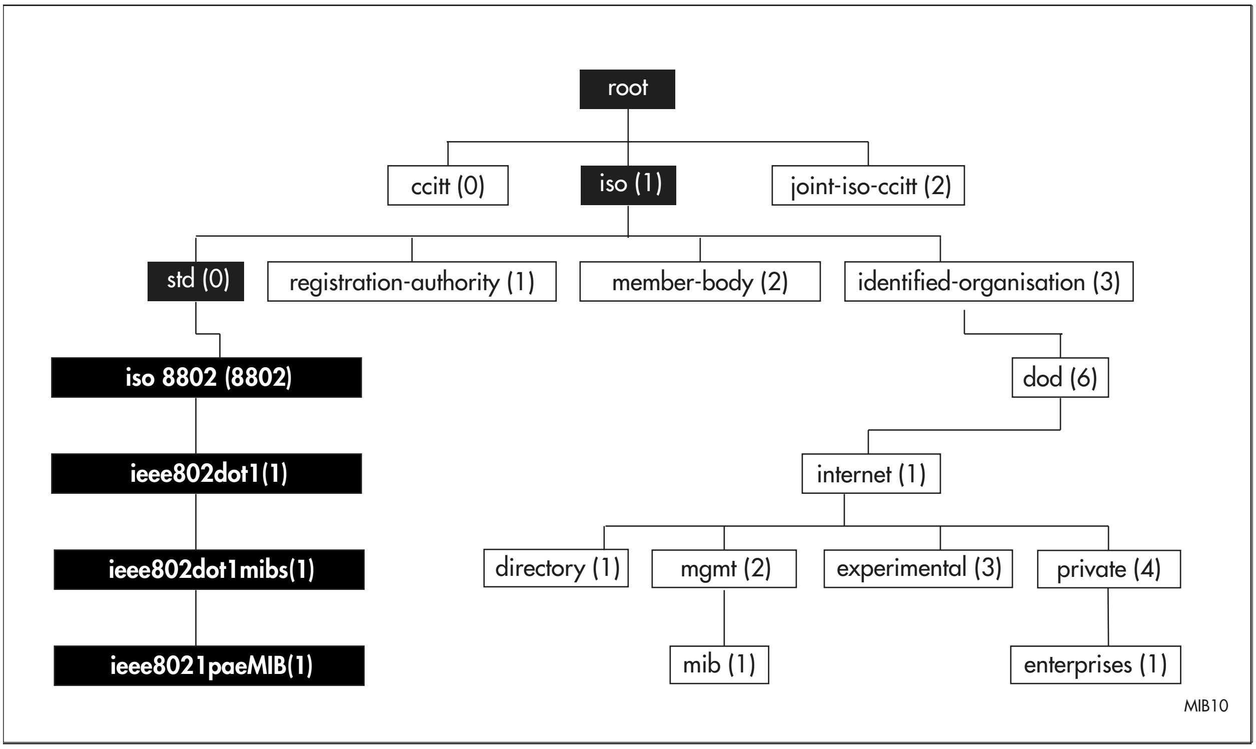
The Internet Draft Definitions for Port Access Control (IEEE Standard 802.1X) MIB (draft-ietf-bridge-8021x-00.txt) defines a portion of the Management Information Base (MIB) for managing the operation of Port Access Control, based on the specification contained in Clause 8 and Clause 9 of the IEEE 802.1X standard. This clause includes a MIB module that is SNMPv2 SMI compliant. Some of the descriptions of the MIB in this section are taken from the Internet Draft.

This MIB is implemented on the AR410 only.

This standard defines a mechanism for Port-based network access control that makes use of the physical access characteristics of IEEE 802 LAN infrastructures in order to provide a means of authenticating and authorizing devices attached to a LAN port that has point-to-point connection characteristics, and of preventing access to that port in cases in which the authentication and authorization process fails.

Objects in this subtree reside in the { iso std(0) ISO 8802(8802) IEEE Standard 802dot1(1) IEEE Standard 802dot1mibs(1) } sub-tree (Figure B-8 on page B-35) and have the object identifier prefix *ieee8021paeMIB* ({ ieee802dot1mibs 1 }).

Figure B-8: The ieee8021paeMIB subtree in the iso MIB



The MIB is organised into three logical groups:

- The *PAE System Group* provides management functionality that is not specific to the operation of either of the two PAE roles (Supplicant and Authenticator). A means of enabling and disabling the operation of Port Access Control for the entire system is provided, plus a per-Port indication of the protocol version supported and the PAE roles supported by the port. As it is not mandatory for all Ports of a System to support PAE functionality, there may be Port entries that indicate Ports that support neither Supplicant nor Authenticator functionality.

- The *PAE Authenticator Group* provides, for each Port of a System, the functionality necessary to allow configuration of the operation of the Authenticator PAE, recording and retrieving statistical information relating to the operation of the Authenticator PAE, and recording and retrieving information relating to a session (i.e., the period of time between consecutive authentications on the Port).
- The *PAE Supplicant Group* provides, for each Port of a System, the functionality necessary to allow configuration of the operation of the Supplicant PAE, and recording and retrieving statistical information relating to the operation of the Authenticator PAE.

Implementation

In the PAE System Group, the router implements:

- *dot1xPaePortTable*

In the PAE Authenticator Group, the router implements:

- *dot1xAuthConfigTable*
- *dot1xAuthStatsTable*

and does not implement the following tables.

- *dot1xAuthDiagTable*
- *dot1xAuthSessionStatsTable*

In the PAE Supplicant Group, the router implements:

- *dot1xSuppConfigTable*
- *dot1xSuppStatsTable*

All tables in this MIB are indexed by port number.

Table B-9: Port Access Control MIB implementation variations

Object Name	Object ID	Access	Implementation
dot1xAuthAdminControlledDirections	{ 1.0.8802.1.1.1.1.2.1.1.3 }	Read-write	The current value of the administrative controlled directions parameter for the Port. Read-only.
dot1xAuthKeyTxEnabled	{ 1.0.8802.1.1.1.1.2.1.1.14 }	Read-write	The value of the *keyTransmissionEnabled* constant currently in use by the Authenticator PAE state machine. Read-only.

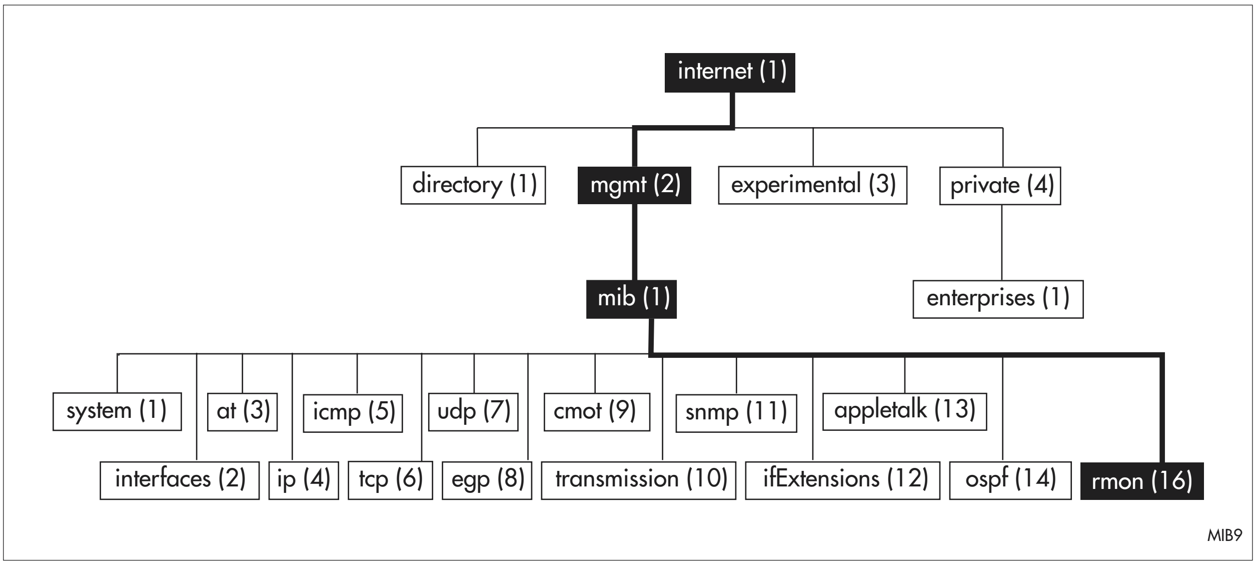
The router also implements a Port Authentication MIB in the Allied Telesyn Enterprise MIB for systems that use a multi-supplicant configuration (see “The Port Authentication Group” on page B-19).

RMON MIB

RFC1757, "*Remote Network Monitoring Management Information Base*" defines a portion of the Management Information Base (MIB) for managing remote network monitoring devices.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-9 on page B-37), and have the object identifier *rmon* ({ mib-2 16 }). These objects are used when the particular media being used to realize an interface is an Ethernet interface.

Figure B-9: RMON in the sub-tree of the Internet-standard Management Information Base (MIB)



The RMON MIB is composed of the following nine groups:

- The Statistics group contains objects that collect statistics measured by the probe for each monitored Ethernet interface on the device. Statistics include the total number of packets received, the size distribution of the packets, types and number of bad packets, and packet collisions.
- The History group contains objects that store periodic statistical samples from the Ethernet network.
- The Alarm group contains objects that periodically collect statistical samples of variables from the probe and compare them to previously configured thresholds, to decide if an event is generated. A hysteresis mechanism is used to limit the number of events generated.
- The Host group contains objects that discover hosts on the network by keeping a list of source and destination MAC addresses seen in good packets. It stores statistics for each host on the network.
- The HostTopN group contains objects that describe hosts that top a list ordered by one of their statistics. Available statistics are samples of one of their base statistics over an interval specified by the management station.
- The Matrix group contains objects that store statistics for conversations between sets of two addresses. It creates a new entry in its tables for each new conversation detected.
- The Filter group contains objects that allow the creation of an arbitrary logical expression with which to capture packets. This data stream may be captured or used to generate events.

- The Packet Capture group contains objects that allow packets to be captured upon a filter match.
- The Event group contains objects that control the generation and notification of events, and whether a log is to be created.

Implementation

The router implements the Statistics, History, Alarm, and Event groups of the RMON MIB for the ETH interface only on the AR410, and for the ETH interfaces and switch ports on the AR440S, AR441S and AR450S.

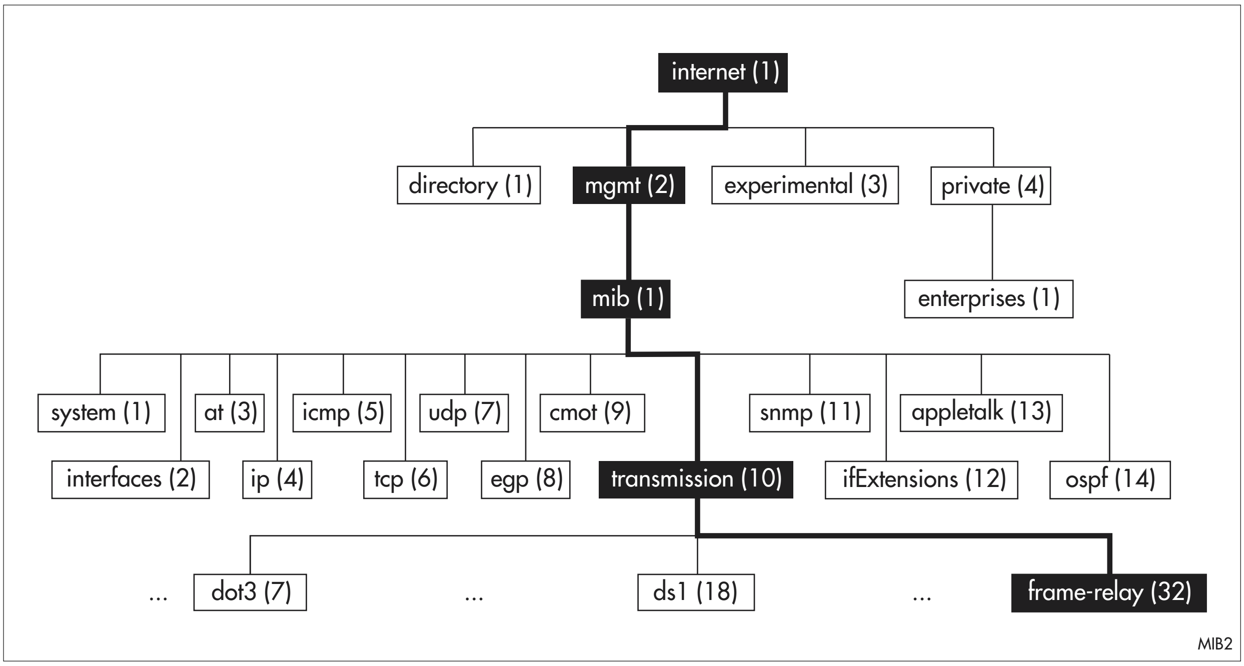
The RMON MIB gathers statistics while SNMP is enabled (Chapter 38, Simple Network Management Protocol (SNMP)).

Frame Relay DTE MIB

RFC 1315, "*Management Information Base for Frame Relay DTEs*" defines a portion of the Management Information Base (MIB) for managing Frame Relay.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-10 on page B-39), under the Transmission Group defined in MIB-II and have the object identifier prefix *frame-relay* ({ transmission 32 }).

Figure B-10: The Frame Relay DTE sub-tree of the Internet-standard Management Information Base (MIB)



For the purposes of understanding this MIB, Frame Relay is viewed as a multi-access media, not as a group of point-to-point connections. This model proposes that Frame Relay is a single interface to the network (physical connection) with many destinations or neighbours (virtual connections). This view enables a network manager the ability to group all virtual connections with their corresponding physical connection thereby allowing simpler diagnostics and trouble shooting.

During normal operation, Frame Relay virtual circuits are added, deleted, and change availability. The occurrence of such changes is of interest to the network manager and therefore, one trap is defined, intended to be corollary to the SNMP "Link Up" and "Link Down" traps.

The objects in the Frame relay DTE MIB are organised into three logical groups:

- The *Data Link Connection Management Interface (DLCMI)* group contains objects that describe the DLC Management Interface on the entity.
- The *Circuit Table* contains objects that describe the use of the Date Link Connection Identifiers (DLCIs) attached to each Frame Relay Interface on the entity. DLCIs enumerate virtual circuits (permanent or dynamic) which are layered onto the underlying circuit, represented by ifEntry. Therefore, each of the entries in the Standard MIB's Interface Table with an IfType of Frame Relay represents a Q.922 interface. Zero or more virtual circuits are layered onto this interface and provide interconnection with various remote destinations. Each such virtual circuit is represented by an entry in the circuit table.

- The *Error Table* contains objects that describe errors encountered on each Frame Relay Interface on the entity.

Implementation

All router models that have synchronous or ISDN interfaces and can support Frame Relay implement all groups in the Frame Relay DTE MIB. However, the implementation of some objects differs from RFC 1315 (Table B-10 on page B-40). In particular, some read-write objects are implemented as read-only and traps are not implemented.

Table B-10: Frame Relay DTE MIB implementation variations

Object Name	Object ID	Access	Implementation
frDlcmiState	{ 1.3.6.1.2.1.10.32.1.1.2 }	Read-write	Read-only.
frDlcmiAddress	{ 1.3.6.1.2.1.10.32.1.1.3 }	Read-write	Read-only.
frDlcmiAddressLen	{ 1.3.6.1.2.1.10.32.1.1.4 }	Read-write	Read-only.
frDlcmiPollingInterval	{ 1.3.6.1.2.1.10.32.1.1.5 }	Read-write	Read-only.
frDlcmiFullEnquiryInterval	{ 1.3.6.1.2.1.10.32.1.1.6 }	Read-write	Read-only.
frDlcmiErrorThreshold	{ 1.3.6.1.2.1.10.32.1.1.7 }	Read-write	Read-only.
frDlcmiMonitoredEvents	{ 1.3.6.1.2.1.10.32.1.1.8 }	Read-write	Read-only.
frDlcmiMaxSupportedVCs	{ 1.3.6.1.2.1.10.32.1.1.9 }	Read-write	Read-only.
frDlcmiMulticast	{ 1.3.6.1.2.1.10.32.1.1.10 }	Read-write	Read-only.
frCircuitState	{ 1.3.6.1.2.1.10.32.2.1.3 }	Read-write	Read-only.
frCircuitCommittedBurst	{ 1.3.6.1.2.1.10.32.2.1.12 }	Read-write	Read-only.
frCircuitExcessBurst	{ 1.3.6.1.2.1.10.32.2.1.13 }	Read-write	Read-only.
frCircuitThroughput	{ 1.3.6.1.2.1.10.32.2.1.14 }	Read-write	Read-only.
frTrapState	{ 1.3.6.1.2.1.10.32.4.1 }	Read-write	Not implemented.
frDLCIStatusChange	-	-	Not implemented.

DS1, E1, DS2 and E2 Interface Types MIB

RFC 2495, *Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types*, defines a portion of the Management Information Base (MIB) for managing DS1, E1, DS2 and E2 interfaces.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-11 on page B-42), under the Transmission Group defined in MIB-II and have the object identifier *ds1* ({ transmission 18 }). These objects are used when the particular media being used to realise an interface is a DS1/E1/DS2/E2 interface. At present, this applies to *ifType* objects with the value *ds1 (18)*.

The definitions in this MIB are based on the AT&T T-1 Superframe format, the Extended Superframe (ESF) formats which conform to ANSI specifications, and the CCITT Recommendations, referred to as E1.

The various DS1 and E1 line disciplines are similar enough that separate MIBs are unwarranted, although there are some differences. For example, Loss of Frame is defined more rigorously in the ESF specification than in the D4 specification, but it is defined in both. Therefore, interface types *e1(19)* and *g703at2mb(67)* have been obsoleted. Where it is necessary to distinguish between the flavours of E1 with and without CRC, E1-CRC denotes the flavour with CRC and E1-noCRC denotes the flavour without CRC.

The MIB is organised into five logical groups:

- The *DS1 Near End Group* contains objects that describe the configuration and performance statistics of the near end of the DS1 link.
- The *DS1 Far End Group* contains objects that describe the configuration and performance statistics of the far end of the DS1 link.
- The *DS1 Fractional Table* contains objects that describe which DS1 channels associated with a CSU are being used to support a logical interface, i.e., an entry in the interfaces table from the Internet-standard MIB.
- The *DS1 Trap Group* contains a trap for line status changes.
- The *Conformance Group* contains objects that describe compliance statements and mandatory object groups.

Implementation

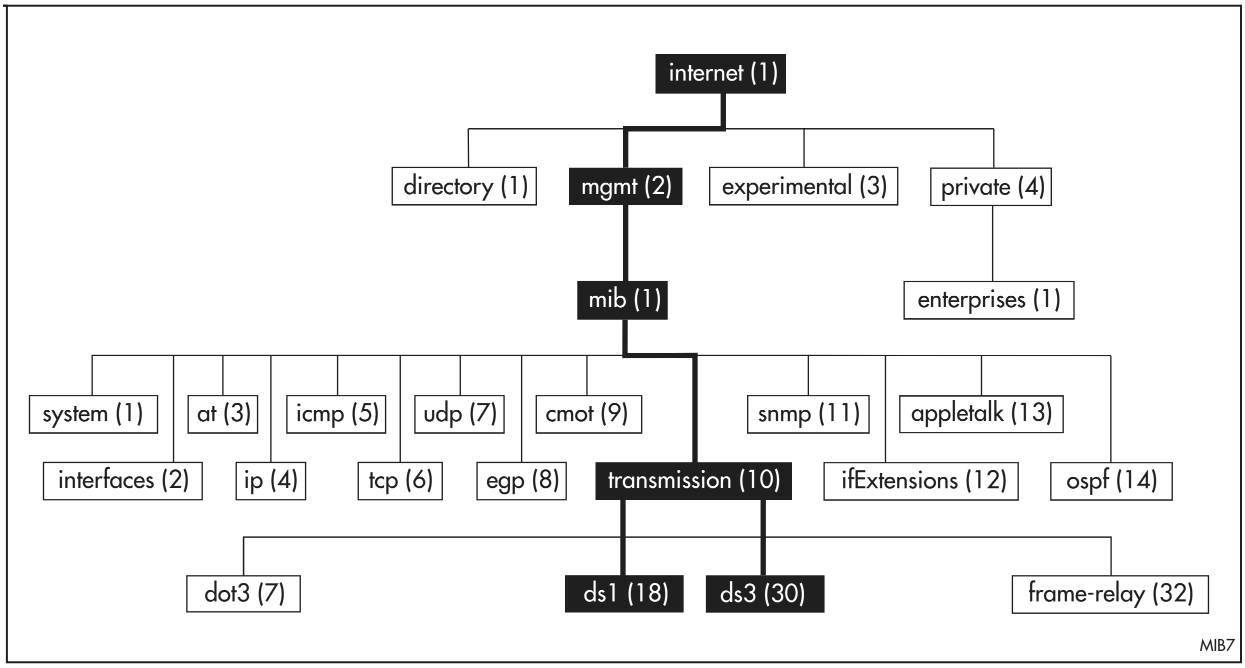
All router models implement the following objects and groups in the DS1/E1/DS2/E2 Interface Types MIB:

- All objects in the *DS1 Near End Group*, except *dsx1ChanMappingTable*. All objects in *dsx1ConfigTable* are implemented as read-only.
- All objects in the *DS1 Far End Group*.

The following objects and groups in the DS1/E1/DS2/E2 Interface Types MIB are not implemented:

- All objects in the *DS1 Fractional Table*.
- All traps in the *DS1 Trap Group*.
- All objects in the *Conformance Group*.

Figure B-11: The DS1/E1/DS2/E2 Interface Types sub-tree of the Internet-standard Management Information Base (MIB)



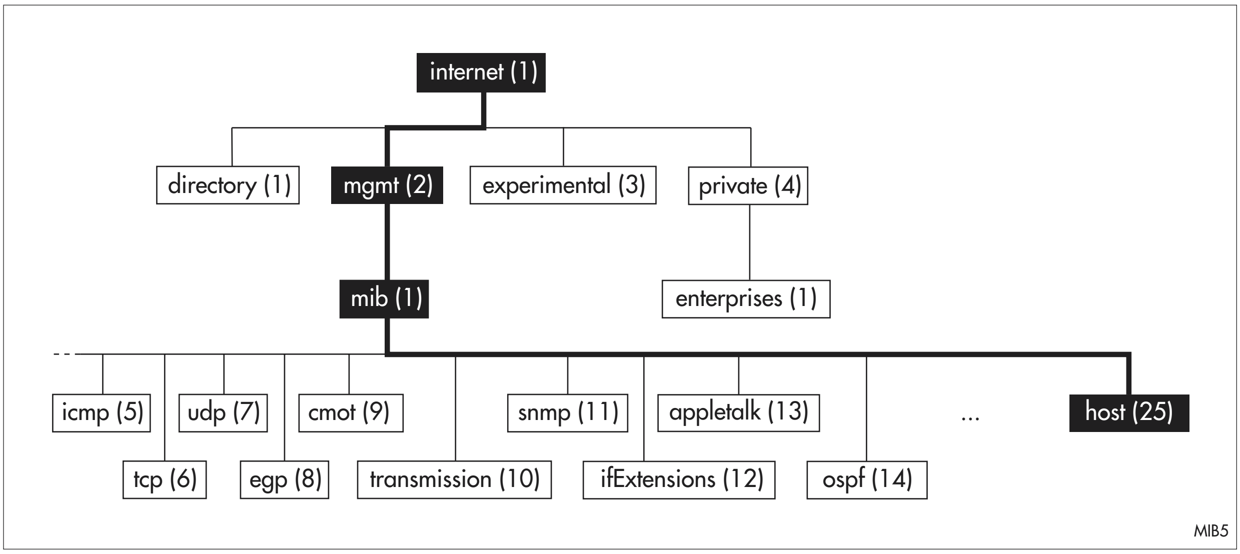
Host Resources MIB

RFC 1514, *Host Resources MIB*, defines a portion of the Management Information Base (MIB) for managing host systems.

The term host is construed to mean any computer that communicates with other similar computers attached to the internet and that is directly used by one or more human beings. Although this MIB does not necessarily apply to devices whose primary function is communications services (such as terminal servers, routers, bridges, monitoring equipment), this relevance is not explicitly precluded. This MIB instruments attributes common to all internet hosts including, for example, both personal computers and systems that run variants of Unix.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-12 on page B-43) and have the object identifier prefix *host* ({ mib-2 25 }).

Figure B-12: The Host Resources sub-tree of the Internet-standard Management Information Base (MIB)



The MIB is organised into six groups:

- The *Host Resources System* group contains objects that describe general system configuration parameters.
- The *Host Resources Storage* group contains objects that describe the logical storage areas on the host.
- The *Host Resources Device* group contains objects that describe the devices on the host.
- The *Host Resources Running Software* group contains objects that describe the software that is running or loaded into physical or virtual memory in preparation for running, including the host's operating system, device drivers, and applications.
- The *Host Resources Running Software Performance* group contains objects that describe performance metrics for each entry in the *hrSWRunTable*.
- The *Host Resources Installed Software* group contains objects that describe each piece of software installed in long-term storage (e.g. a disk drive) locally on this host. This does not include software loadable remotely from a network server.

Implementation

All router models implement the following groups in the Host Resources MIB:

- The *Host Resources System* group.
- The *Host Resources Storage* group.
- The *Host Resources Device* group.

However, the implementation of some objects differs from RFC 1514 (Table B-11 on page B-44).

The following groups in the Host Resources MIB are not implemented because they are not meaningful in the context of the router:

- The *Host Resources Running Software* group
- The *Host Resources Running Software Performance* group
- The *Host Resources Installed Software* group

Table B-11: Host Resources MIB implementation variations

Object Name	Object ID	Access	Implementation
hrSystemDate	{ 1.3.6.1.2.1.25.1.2 }	Read-write	Read-only.
hrSystemInitialLoadDevice	{ 1.3.6.1.2.1.25.1.3 }	Read-write	The index of the *hrDeviceEntry* for the device (NVS or FLASH) containing the system boot script. Read-only.
hrSystemInitialLoadParameters	{ 1.3.6.1.2.1.25.1.4 }	Read-write	The name of the system boot script (as set by the SET CONFIG command). Read-only.
hrSystemNumUsers	{ 1.3.6.1.2.1.25.1.5 }	Read-only	The number of active user (Telnet and asynchronous) sessions that have been authenticated via the USER module.
hrSystemProcesses	{ 1.3.6.1.2.1.25.1.6 }	Read-only	Always returns the value 1.
hrSystemMaxProcesses	{ 1.3.6.1.2.1.25.1.7 }	Read-only	Always returns the value 1.
hrStorageAllocationUnits	{ 1.3.6.1.2.1.25.2.3.1.4 }	Read-only	Returns the value 2048 for RAM buffers, or 1 for NVS and FLASH.
hrStorageSize	{ 1.3.6.1.2.1.25.2.3.1.5 }	Read-write	Read-only.
hrDeviceType	{ 1.3.6.1.2.1.25.3.2.1.2 }	Read-only	Table B-12 on page B-45 lists the devices supported by the router.
hrDeviceID	{ 1.3.6.1.2.1.25.3.2.1.4 }	Read-only	Object identifier values are from the Objects Group of the Enterprise MIB on page B-7.
hrNetworkTable	{ 1.3.6.1.2.1.25.3.4 }	-	Not implemented.
hrPrinterTable	{ 1.3.6.1.2.1.25.3.5 }	-	Not implemented. The router provides print services (e.g. LPD) but it is not concerned with issues such as printer diagnosis which are required to implement this portion of the MIB.
hrDiskStorageTable	{ 1.3.6.1.2.1.25.3.6 }	-	Not implemented because the router does not have any entries in the device table of type *hrDeviceDiskStorage*. FLASH and NVS are represented as raw non-volatile memory.
hrFSTable	{ 1.3.6.1.2.1.25.3.8 }	-	Not implemented because the router does not support any file systems close enough in semantics to those catered for by this MIB.

Table B-11: Host Resources MIB implementation variations (continued)

Object Name	Object ID	Access	Implementation
Host Resources Running Software Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.
Host Resources Running Software Performance Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.
Host Resources Installed Software Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.

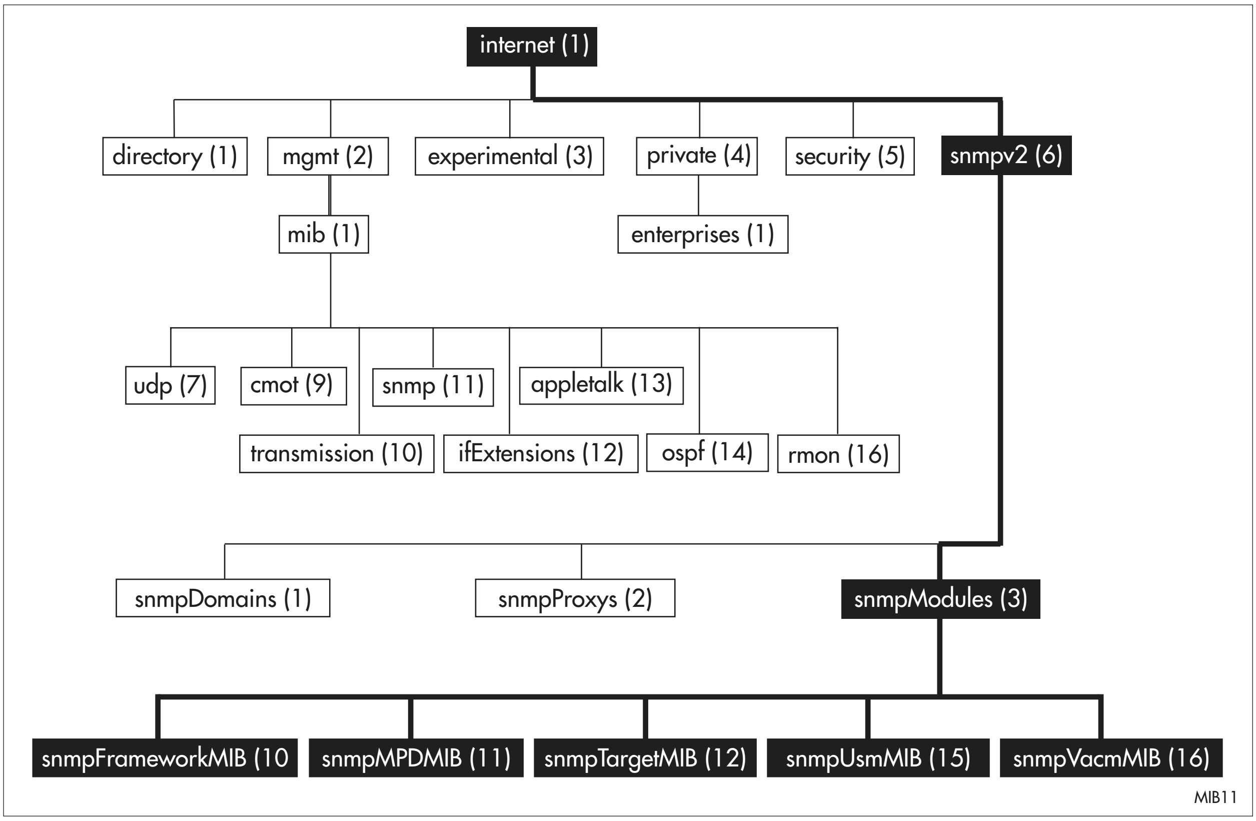
Table B-12: Host Resources MIB device types supported by the router

hrDeviceType	Router Device
hrDeviceProcessor	Processor
hrDeviceNetwork	LAN/WAN network port
hrDeviceCoprocessor	MAC card
hrDeviceSerialPort	Asynchronous ports
hrDeviceClock	RTC
hrDeviceVolatileMemory	RAM, FSRAM, CAM
hrDeviceNonVolatileMemory	NVS, flash

SNMP v3 MIBs

Sub-trees of the snmpModules MIB branch (OID: 1.3.6.1.6.3) are shown in figure Figure B-9 below. SNMPv3 modules sub-tree of the Internet-standard MIB object tree are shown below.

Figure B-13: SNMPv3 subnet-tree of the Internet-standard Management Information Base (MIB)



This branch comprises the following MIBs

- *SNMP-FRAMEWORK-MIB* - comprises the snmpEngine, which contains information about managed objects used to instrument SNMP entities. This MIB is defined RFC 3411.
- *Message Processing and Dispatching MIB* - contains objects that define SNMP message processing and dispatching. It consists of the table snmpMPDStats, which provide information about SNMP message statistics. This MIB is defined in RFC 3412.
- *Management Target MIB* - is intended to provide a general-purpose mechanism for specifying transport addresses and parameters for SNMP messages generated by an SNMP entity. This MIB is defined in RFC 3413.

- *User-based Security MIB Module* - contains objects for defining the User-based Security Model (USM) for SNMPv3, and for remotely monitoring and managing the configuration parameters for this security model. This MIB is defined in RFC 3414.
- *View-based Access Control MIB Module* - contains objects for defining SNMP View-based Access Control Model (VACM) and for remotely managing the configuration parameters for the VACM. This MIB is defined in RFC 3415.

Implementation

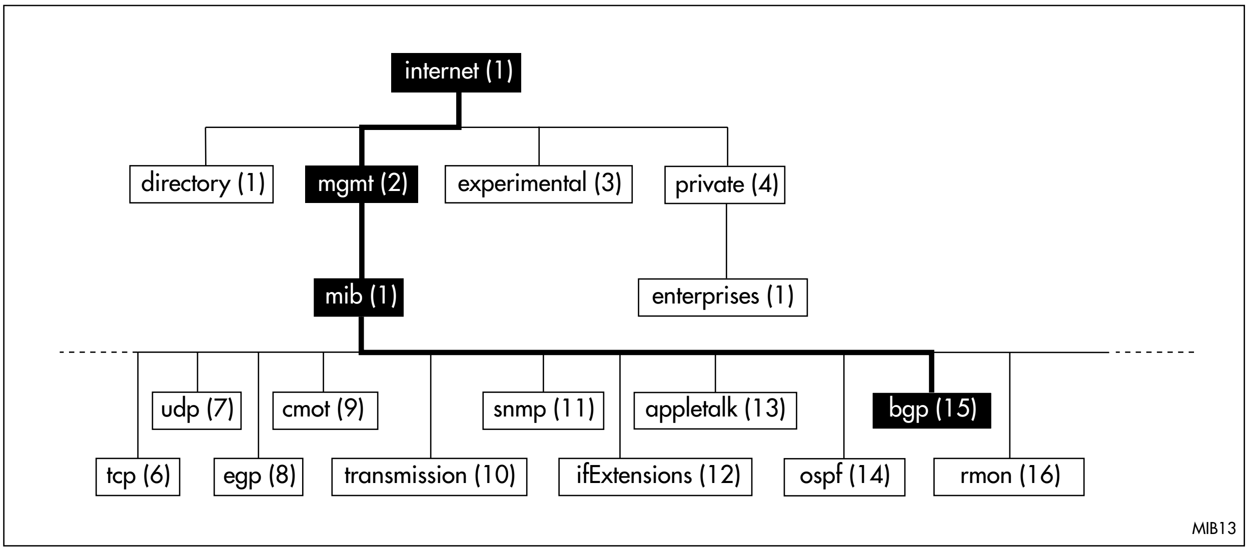
All router models implement all objects in the SNMPv3 MIBs.

Border Gateway Protocol v4 (BGP-4) MIB

RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*, defines a portion of the Management Information Base (MIB) for managing the Border Gateway Protocol Version 4 or lower.

Objects defined in this MIB reside in the mib(1) sub-tree (Figure B-14 on page B-48) and have the object identifier prefix *bgp* ({ mib-2 15 }).

Figure B-14: The BGP-4 sub-tree of the Internet-standard Management Information Base (MIB)



The BGP-4 MIB contains the following objects:

- The *bgpVersion* object contains information about the supported BGP protocol version numbers.
- The *bgpLocalAs* object contains the local autonomous system number.
- The BGP peer table (*bgpPeerTable*) contains objects that describe the state and current activity of connections with BGP peers.
- The *bgpIdentifier* object contains the BGP Identifier (IP address) of the local system.
- The BGP Received Path Attribute Table (*bgpRcvdPathAttrTable*) contains objects that describe the attributes of paths to destination networks received from all peers running BGP version 3 or less. This table has been deprecated.
- The BGP-4 Received Path Attribute Table (*bgp4PathAttrTable*) contains objects that describe the attributes of paths to destination networks received from all BGP-4 peers. The actual attributes used in determining a route are a subset of the received attribute tables after local routing policy has been applied.

- The *bgpEstablished* trap is generated when the BGP FSM enters an established state, and contains objects that describe the state of the BGP peer connection and the last error code and subcode seen by the local peer on the connection.
- The *bgpBackwardTransition* trap is generated when the BGP FSM moves from a higher numbered state to a lower numbered state, and contains objects that describe the state of the BGP peer connection and the last error code and subcode seen by the local peer on the connection.

Implementation

All router models implement all objects except the BGP Received Path Attribute Table (bgpRcvdPathAttrTable) which has been deprecated and does not apply to BGP version 4.