

Appendix C

SNMP MIBs

Introduction	C-3
Allied Telesis Enterprise MIB	C-4
Products Subtree	C-5
AT Router Subtree	C-9
Objects Group	C-9
traps Group	C-16
sysinfo Group	C-16
arInterfaces Group	C-17
Modules Group	C-18
Ethernet Group	C-18
Flash Group	C-18
TTY Group	C-19
ISDN Call Control Group	C-19
BRI Group	C-20
PRI Group	C-20
Loader Group	C-20
Install Group	C-20
Trigger Group	C-21
File Group	C-21
Ping Group	C-22
DHCP Group	C-22
Firewall Group	C-22
Switch Group	C-23
PIM4 Group	C-24
QoS Group	C-24
Load Balancer Group	C-24
DS3 Group	C-25
Port Authentication Group	C-25
Stack Group	C-26
Software QoS Group	C-27
MIB-II MIB	C-29
Implementation	C-30
Interfaces Group MIB	C-31
Implementation	C-32
IP Forwarding Table MIB	C-33
Implementation	C-33
Ethernet-like Interface Types MIB	C-34
Implementation	C-34
Medium Attachment Unit (MAU) MIB	C-35
Implementation	C-36
Bridge MIB	C-37

Implementation	C-38
Virtual Bridge (VLAN) MIB	C-39
Implementation	C-39
Link Aggregation MIB	C-41
Implementation	C-42
Port Access Control MIB	C-43
Implementation	C-44
RMON MIB	C-45
Implementation	C-46
Frame Relay DTE MIB	C-47
Implementation	C-48
DS1, E1, DS2 and E2 Interface Types MIB	C-49
Implementation	C-50
DS3/E3 Interface Types MIB	C-51
Implementation	C-51
Host Resources MIB	C-52
Implementation	C-53
SNMP v3 MIBs	C-55
Implementation	C-55
Border Gateway Protocol v4 (BGP-4) MIB	C-56
Implementation	C-57
Link Layer Discovery Protocol (LLDP) MIB	C-58
Implementation	C-58
State and Topology Change Traps	C-59

Introduction

This appendix describes the Management Information Bases (MIBs) and managed objects implemented by the Rapier i switch, including any variations from the RFC standards. [Table C-1](#) lists the MIBs supported by the switch. For information about configuring SNMP on the switch see [Chapter 55, Simple Network Management Protocol \(SNMP\)](#).

Table C-1: MIBs supported by the switch

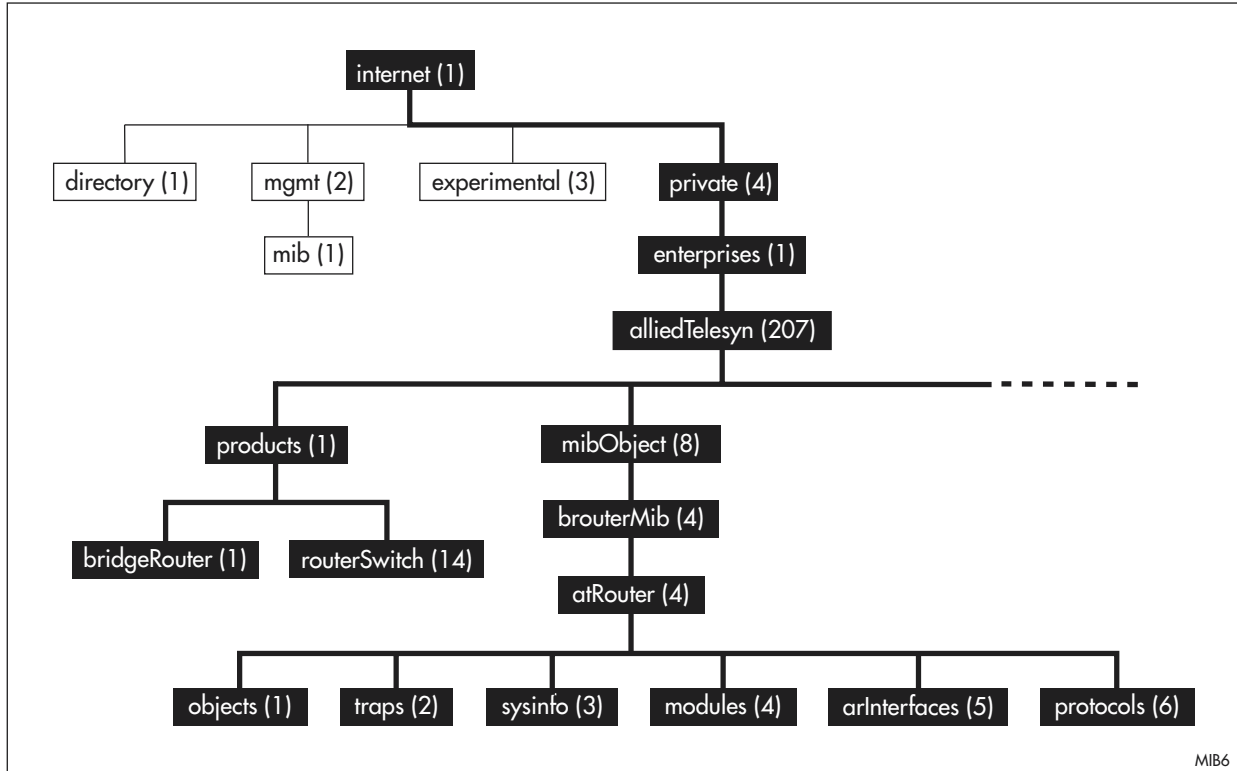
SNMP MIB	Description
Allied Telesis Enterprise MIB	Objects in the Allied Telesis Enterprise MIB for managing the switch.
MIB-II	The core set of objects for TCP/IP internets.
Interface Extensions MIB	Extensions to the interfaces table of MIB-II.
IP Forwarding Table MIB	Objects for managing Classless Inter-Domain Routing (CIDR).
Ethernet-like Interface Types MIB	Objects for managing Ethernet-like interfaces.
Medium Attachment Unit (MAU) MIB	Objects for managing medium attachment units.
Bridge MIB	Objects for managing MAC bridges based on the IEEE 802.1D-1990 standard.
Virtual Bridge (VLAN) MIB	Objects for managing VLANs based on the IEEE 802.1Q-1998 standard.
Link Aggregation MIB	Objects for managing link aggregation based on the IEEE 802.3-2002 standard.
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	Objects for managing Frame Relay.
DS1, E1, DS2 and E2 Interface Types MIB	Objects for managing DS1, E1, DS2 and E2 interfaces.
DS3/E3 Interface Types MIB	Objects for managing DS3 and E3 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).
Link Layer Discovery Protocol (LLDP) MIB	Objects for managing the Link Layer Discovery Protocol (LLDP).
State and Topology Change Traps	State and topology change traps for OSPF, VRRP and MSTP.

Allied Telesis Enterprise MIB

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

Objects defined in this MIB reside in the private(4) subtree ([Figure C-1](#)) and have the object identifier prefix alliedTelesyn (1 enterprises 207).

Figure C-1: The Allied Telesis Enterprise MIB subtree of the Internet-standard Management Information Base (MIB)



Products Subtree

The Products subtree contains a set of object identifiers for Allied Telesis products (Table C-2). Objects have the identifier prefix of products ({ alliedTelesis 1 }). Within this subtree:

- objects with the identifier prefix bridgeRouter ({ products 1 }) are routers
- objects with the identifier prefix routerSwitch ({ products 14 }) are switches with routing functionality

Table C-2: Object identifiers for Allied Telesis router and switch 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

Table C-2: Object identifiers for Allied Telesis router and switch products (Continued)

Object	Object Identifier	Description
at-AR440Router	{ bridgeRouter 59 }	AT-AR440S ADSL Router (Annex A)
at-AR441Router	{ bridgeRouter 60 }	AT-AR441S ADSL Router (Annex B)
at-AR442Router	{ bridgeRouter 61 }	AT-AR442S SHDSL Router
at-AR443Router	{ bridgeRouter 62 }	AT-AR443S ADSL2+ Router
at-AR444Router	{ bridgeRouter 63 }	—Reserved—
at-AR420Router	{ bridgeRouter 64 }	—Reserved—
at-AR415Router	{ bridgeRouter 71 }	AT-AR415S Secure Modular Router
at-AR415RouterDC	{ bridgeRouter 72 }	AT-AR415S Secure Modular Router (DC power)
at-AR550Router	{ bridgeRouter 73 }	AT-AR550 Broadband & ISDN Router
at-AR551Router	{ bridgeRouter 74 }	AT-AR750S Secure VPN Router
at-AR552Router	{ bridgeRouter 75 }	—Reserved—
at-AR550RouterDP	{ bridgeRouter 76 }	AT-AR550S Broadband & ISDN Router with dual PSU
at-AR570Router	{ bridgeRouter 78 }	AT-AR570 Secure VPN Router
at-AR770Router	{ bridgeRouter 79 }	AT-AR770S Secure VPN Router
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)
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)

Table C-2: Object identifiers for Allied Telesis router and switch products (Continued)

Object	Object Identifier	Description
at-Rapier16fSCi	{ routerSwitch 19 }	Rapier 16F-FX/SCi switch, 16 x 100BASE-FX ports (SC fibre connectors)
at-Rapier24i	{ routerSwitch 20 }	Rapier 24i switch, 24 x 10BASE-T/100BASE-TX ports (RJ-45 connectors)
at-Rapier48i	{ routerSwitch 21 }	Rapier 48i 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)
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

Table C-2: Object identifiers for Allied Telesis router and switch products (Continued)

Object	Object Identifier	Description
at-8948i	{ routerSwitch 45 }	AT-8948i 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 24i 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
at-9924TEMC	{ routerSwitch 53 }	—Reserved—
at-8724MLB	{ routerSwitch 55 }	—Reserved—
at-8624PoE	{ routerSwitch 56 }	AT-8624PoE switch, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2 port 1000BASE-X (GBIC slots)
at-9924Ts	{ routerSwitch 57 }	AT-9924Ts Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays
at-86482SP	{ routerSwitch 58 }	AT-8648/2SP Advanced Fast Ethernet Switch, 48 x 10BASE-T/100BASE-TX ports (RJ 45 connectors), 2 x SFP Gigabit ports or 2 x 10BASE-T/ 100BASE-T/1000BASE-T/ ports (RJ 45 connectors)
at-9924Ti	{ routerSwitch 59 }	AT-9924Ti Multi-layer IPv4 Gigabit Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 4 SFP Gigabit uplink ports
at-9924SPi	{ routerSwitch 60 }	AT-9924SPi Multi-layer IPv4 Gigabit Switch, 24 SFP Gigabit uplink ports
at-9924Tsi	{ routerSwitch 61 }	AT-9924Tsi Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays
at-9924SPsi	{ routerSwitch 62 }	AT-9924SPsi Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays
at-8948i-N	{ routerSwitch 63 }	AT-8948i Multi-layer Fast Ethernet Switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors), 4 SFP Gigabit uplink ports, NEBS compliant
at-9924Tsi-N	{ routerSwitch 64 }	AT-9924Tsi Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays, NEBS compliant
at-Rapier48w	{ routerSwitch 65 }	Rapier 48w switch, 48 x 10BASE-T/100BASE-TX ports (RJ-45 connectors), nSM bay, NEBs compliant

AT Router Subtree

The AT Router subtree contains a set of objects for managing multiprotocol routers and switches. Objects have the identifier prefix of atRouter ({ alliedTelesis(207) mibObject(8) brouterMib(4) 4 }). Objects are arranged into six groups ([Table C-3](#)).

Table C-3: Object groups in the AT Router subtree of the Allied Telesis Enterprise MIB

Group	Object Identifier (OID)
objects	{ atRouter 1 }
traps	{ atRouter 2 }
sysinfo	{ atRouter 3 }
modules	{ atRouter 4 }
arInterfaces	{ atRouter 5 }
protocols	{ atRouter 6 }
atAgents	{ atRouter 7 }

Objects Group

The Objects Group contains four sets of object identifiers for:

- boards ({ objects 1 }), listed in [Table C-4](#). These object identifiers are for use with the hrDeviceID object in the Host Resources MIB ([Table C-13 on page C-53](#)).
- releases ({ objects 2 }).
- interface types ({ objects 3 }), listed in [Table C-5 on page C-15](#).
- chip sets ({ objects 4 }), listed in [Table C-6 on page C-15](#).

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

Object	Object Identifier	Description
boards	{ objects 1 }	
pprlcmAr023	{ boards 39 }	AT-AR023 SYN1 PIC card
pprlcmAr021s	{ boards 40 }	AT-AR021s BRI1 (S/T interface) PIC card
pprlcmAr022	{ boards 41 }	AT-AR022 ETH1 PIC card
pprlcmAr025	{ boards 45 }	AT-AR025 E1 PRI1 PIC card
pprlcmAr024	{ 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)
pprlcmAr021u	{ boards 59 }	AT-AR021u BRI1 (U interface) PIC card
pprAr720	{ boards 63 }	AT-AR720 (2 PIC card slots)

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
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)
pprlcmAr020	{ boards 76 }	AT-AR020 T1 PRI1 PIC card
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)

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
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)
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 24i 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)

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
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)
pprlcmAR026	{ 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)
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)

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
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)
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)
pprlcmAr027	{ 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)
ppr8948i	{ boards 203 }	AT-8948i (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)
pprlcmAr021v2s	{ boards 206 }	AT-AR021v2s BRI1 (S/T interface) PIC card
pprA50	{ boards 207 }	—Reserved—
pprA51	{ boards 208 }	—Reserved—
pprA52	{ boards 209 }	—Reserved—
pprA53	{ boards 210 }	—Reserved—
pprFanA01	{ boards 212 }	—Reserved—
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
pprSb24RJ	{ boards 216 }	AT-SB4412 24-port 1000BASE-T (RJ45) Gigabit Ethernet Line Card
pprSb1XFP	{ boards 217 }	AT-SB4541A 1-port 10GBASE-R (XFP) 10 Gigabit Ethernet Line Card
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
ppr9924TEMC	{ boards 220 }	—Reserved—
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

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
pprAR440	{ boards 227 }	AT-AR440S ADSL Router (Annex A)
pprAR441	{ boards 228 }	AT-AR441S ADSL Router (Annex B)
pprAR442	{ boards 229 }	AT-AR442S SHDSL Router
pprAR443	{ boards 230 }	AT-AR443S ADSL2+ Router
pprAR444	{ boards 231 }	—Reserved—
pprAR420	{ boards 232 }	—Reserved—
pprAt8624T2M	{ boards 239 }	AT-8624T2M Layer 3 Fast Ethernet Switch
pprA46Tx	{ boards 240 }	AT-A46 1000Base-X Uplink
pprAR550	{ boards 241 }	AT-AR550 Broadband & ISDN Router
pprAR551	{ boards 242 }	AT-AR750S Secure VPN Router
pprAR552	{ boards 243 }	—Reserved—
pprC8724MLB	{ boards 248 }	—Reserved—
pprAt86482SP	{ boards 252 }	AT-8648/2SP Advanced Fast Ethernet Switch, 48 x 10BASE-T/ 100BASE-TX ports (RJ 45 connectors), 2 x SFP Gigabit ports, or 2 x 10BASE-T/ 100BASE-T/1000BASE-T/ ports (RJ 45 connectors)
pprAT-8624PoE	{ boards 253 }	AT-8624P0E, 24 10BASE-T/100BASE-TX ports (RJ-45 connectors), 2 port 1000BASE-X (GBIC slots)
pprPwr01RAC	{ boards 254 }	AT-PWR01 AC Power Supply Unit - Reverse Fans
pprFan01R	{ boards 255 }	AT-FAN01 Fan Only Module - Reverse Fan
ppr9924Ts	{ boards 256 }	AT-9924Ts Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays
pprAR570	{ boards 258 }	AT-AR570 High Performance VPN Router
pprPwr02AC	{ boards 264 }	AT-PWR02 AC Power Supply Unit
pprPwr02RAC	{ boards 265 }	AT-PWR02 AC Power Supply Unit - Reverse Fans
pprAtXum10G	{ boards 266 }	AT-A60 Expansion Module, 1 x 10Gbe XFP port
pprAtXum12T	{ boards 267 }	AT-A62 Expansion Module, 12 x 10/100/100BASE-T copper ports (RJ-45 connectors)
pprAtXum12SFP	{ boards 268 }	AT-A61 Expansion Module, 12 x SFP Gigabit ports
pprSb24SFP	{ boards 269 }	AT-SB4442 24-port 1000BASE-X (SFP) Gigabit Ethernet Line Card
pprAR770	{ boards 270 }	AT-AR770S Secure VPN Router
pprAtXum10Gi	{ boards 273 }	XEM-1XP Expansion Module, 1 x 10Gbe XFP port
pprAtXum12SFPi	{ boards 274 }	XEM-12S Expansion Module, 12 x SFP Gigabit ports
pprAtXum12Ti	{ boards 275 }	XEM-12T Expansion Module, 12 x 10/100/100BASE-T copper ports (RJ-45 connectors)
pprAR415S	{ boards 276 }	AT-AR415S Secure Modular Router
pprAR415SDC	{ boards 277 }	AT-AR415S Secure Modular Router (DC power)
pprAR550SDP	{ boards 278 }	AT-AR550S Broadband & ISDN Router with dual PSU
ppr8948iN	{ boards 279 }	AT-8948i-N (4 10/100 Ethernet switch ports, 4 1000BASE-X SFP uplink ports, 1 CompactFlash socket, 1 RS-232 async port), NEBS compliant
pprAtXum12TiN	{ boards 280 }	XEM-12T-N Expansion Module, 12 x 10/100/100BASE-T copper ports (RJ-45 connectors), NEBS compliant

Table C-4: Object identifiers for base CPU and expansion boards (Continued)

Object	Object Identifier	Description
ppr9924TsiN	{ boards 281 }	AT-9924Tsi Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/100BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays, NEBS compliant
pprRapier48w	{ boards 283 }	Rapier 48 (48 10BASE-T/100BASE-T ports, RJ-45 connectors), NSM expansion bay, NEBS compliant

Table C-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 C-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

Table C-6: Object identifiers for chip sets (Continued)

Object	Object Identifier	Description
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

traps Group

The traps Group has the object identifier prefix traps ({ atRouter 2 }), and contains the following generic traps:

- The igmpGeneralQueryNotReceivedEvent trap ({ igmpTraps(1) 1 }) is generated when no general IGMP query message is received on an interface within a user-defined time period, and contains the ifName object for the interface.

sysinfo Group

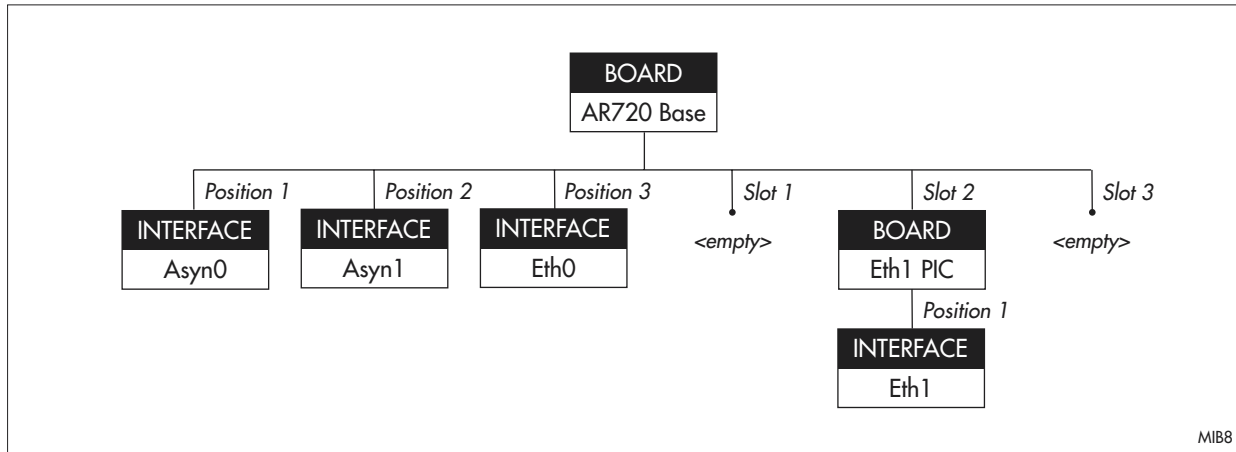
The sysinfo Group has the object identifier prefix sysinfo ({ atRouter 3 }), and contains the following objects that describe generic system information:

- fanAndPs ({ sysinfo 1 }) is a collection of objects and traps for monitoring the fan, power supply, and temperature status.
- restartGroup ({ sysinfo 2 }) is a collection of objects and traps for activating and monitoring restarts.
- 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 and traps for monitoring and managing the temperature status.
- atContactDetails ({ sysinfo 5 }) contains the contact details for Allied Telesis.
- bbrNvs ({ sysinfo 6 }) contains a trap for monitoring the status of the battery backed RAM non-volatile storage.
- memory ({ sysinfo 7 }) is a collection of objects and traps for monitoring memory usage.
- realTimeClockStatus ({ sysinfo 8 }) contains information about the status of the real time clock.
- hostId ({ sysinfo 9 }) sets a host ID for the switch, and enables the port numbering format, *hostid.board.port*, for switch ports.
- envMon ({ sysinfo 10 }) is a collection of additional objects and traps for monitoring and managing fan speeds, voltage rails, temperature sensors, and power supply units.

arInterfaces Group

The arInterfaces Group ({ atRouter 5 }) contains objects that describe the boards, slots and physical interfaces in the switch. A switch 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 switch may be seen as a tree whose nodes are boards and interfaces, and whose links are positions and slots. Figure C-2 illustrates the object tree for an AR720 router with two asynchronous ports and an Ethernet port on the base CPU board, and a single Ethernet PIC in Bay 1.

Figure C-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 switch has hot-swap capability. When the switch 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 switch.
- arBoardTable ({ arInterfaces 2 }) is a table of the boards in the switch, indexed by board index (arBoardIndex). This table describes all of the physical boards present in the switch. 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 switch. 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 switch. This table is indexed by board index (arInterfaceBoardIndex) and a board position (arInterfacePosition), and has an entry for each physical interface on the switch.
- 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 switch.

Modules Group

The Modules Group ({ atRouter 4 }) contains objects that describe software modules not covered by standard MIBs.

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 switch:

- ethIntTable ({ ethernet 1 }) is a table of the Ethernet interfaces on the switch. Each entry is a single Ethernet interface on the switch 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.

Flash Group

The Flash Group has the object identifier prefix flash ({ modules 31 }), and contains a collection of objects that describe Flash memory operations.

The following objects are defined:

- flashFailureTrap ({ flash 0 }) is a trap generated when a flash operation fails, and contains the following objects:
 - flashGetFailure ({ flash 1 }), the number of flash get operations that have failed
 - flashOpenFailure ({ flash 2 }), the number of flash open operations that have failed
 - flashReadFailure ({ flash 3 }), the number of flash read operations that have failed
 - flashCloseFailure ({ flash 4 }), the number of flash close operations that have failed
 - flashCompleteFailure ({ flash 5 }), the number of flash complete operations that have failed
 - flashWriteFailure ({ flash 6 }), the number of flash write operations that have failed
 - flashCreateFailure ({ flash 7 }), the number of flash create operations that have failed
 - flashPutFailure ({ flash 8 }), the number of flash put operations that have failed
 - flashDeleteFailure ({ flash 9 }), the number of flash delete operations that have failed
 - flashCheckFailure ({ flash 10 }), the number of flash check operations that have failed
 - flashEraseFailure ({ flash 11 }), the number of flash erase operations that have failed
 - flashCompactFailure ({ flash 12 }), the number of flash compaction operations that have failed

- flashVerifyFailure ({ flash 13 }), the number of flash verify operations that have failed

TTY Group

The TTY Group has the object identifier prefix tty ({ modules 36 }), and contains a collection of objects and traps for monitoring login failures.

The following objects are defined under the ttyTraps ({ tty 100 }) subtree:

- loginFailureUser ({ ttyTraps 1 }) is the username that generated the login failure.
- loginFailureIPAddress ({ ttyTraps 2 }) is the IP address the failed login attempt originated from.
- loginFailureAttempts ({ ttyTraps 3 }) is the number of failed login attempts.
- The loginFailureTrap trap ({ ttyTraps 11 }) is generated when a user is locked out because the number of consecutive failed login attempts exceeded the maximum allowed, and contains the following objects:
 - loginFailureUser
 - loginFailureIPAddress
 - loginFailureAttempts

ISDN Call Control Group

For switches that support ISDN, the ISDN Call Control Group contains objects that describe ISDN call definitions, active call details and call history on the switch, 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 switch, 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 switch 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 switch 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 switch processes, this table is read-only.

BRI Group

For switches that support BRI, the BRI Group contains objects that describe BRI interfaces on the switch, 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 switch, indexed by `ifIndex`. Each entry is a single BRI interface on the switch 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.

PRI Group

For switches that support PRI, the PRI Group contains objects that describe PRI interfaces on the switch, 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 switch, indexed by `ifIndex`. Each entry is a single PRI interface on the switch 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.

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 switch. 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.

Install Group

The Install Group contains objects for managing the INSTALL module that controls the software release and patch running on the switch. 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 switch. 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 switch. Each entry is a descriptive line that tells of part of the install history of the last switch reboot.
- `configFile` ({ install 3 }) is the name of the file that the switch configures from at boot. If the configuration file name is a zero length string, then no configuration file is defined in the switch.
- `licenceTable` ({ install 4 }) is a table of release licences in the switch. This table contains licences for releases of switch 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 switch 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.
- `configFileExist` ({ install 6 }) returns true(1) if the boot configuration file exists, or false(2) if it does not exist.
- `configFileExistTrap` ({ install(49) installTrap(0) 1 }) is a trap message generated when the switch detects that the boot file is missing. It returns the value of `configFileExist`.
- `currentConfigFile` ({ install 7 }) is the name of the file that the switch is currently configured with. If the configuration file name is a zero length string, then there is no current configuration file defined.

Trigger Group

The Trigger Group contains objects for managing triggers in the switch. 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 switch.
- `triggerTrap` ({ trigger(53) triggerTrap(0) 1 }) is a trap message generated when a trigger is activated. It returns the value of `triggerLastTriggerActivated`.

File Group

The File Group contains objects for managing the file system in the switch. 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 switch'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 switch.

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 ({ ping(58) pingTraps(0) 1 }) is a trap message generated when a ping has completed.

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 switch.
- dhcpTrapVariable ({ dhcp 2 }) is a collection of variables used in trap messages:
 - dhcpRangeExhaustedGateway ({ dhcpTrapVariable 1 }), the gateway address of a DHCP request that could not be completed because the server did not have any free addresses.
 - dhcpRangeExhaustedInterface ({ dhcpTrapVariable 2 }), the IP address of the interface that received a DHCP request that could not be completed because the server did not have any free addresses.
- dhcpRangeExhaustedTrap { dhcp(70) dhcpTraps(0) 1 } is a trap generated when a client requests an IP address and the DHCP server can not satisfy the request because all addresses in the range are already allocated. It returns the objects dhcpRangeExhaustedGateway and dhcpRangeExhaustedInterface.

Firewall Group

For switches that support a firewall, the Firewall Group contains objects describing traps generated by the firewall in the switch. Objects in this group have the object identifier prefix firewall ({ modules 77 }).

The following objects are defined:

- `firewallTrapMessage` ({ `firewall 1` }) 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.
- `firewallAttackTrap` ({ `firewall(77) firewallTraps(0) 1` }) is a trap message generated when the firewall detects an intrusion or attack. It returns the object `firewallTrapMessage`. Firewall trap notifications are enabled using the command:

```
enable firewall notify=snmp
```

- `firewallSessionStatistics` ({ `firewall 2` }) is a collection of objects for monitoring firewall sessions:
 - `totalNumberOfSessions` ({ `firewallSessionStatistics 1` }), the total number of active sessions through the firewall.
 - `numberOfSessionsPerNodeCountingStatus` ({ `firewallSessionStatistics 2` }), the status of session reporting—enabled(1) or disabled(2).
 - `numberOfSessionsPerNodeTable` ({ `firewallSessionStatistics 2` }), a table of node IP addresses (`nodeIpAddress`) and the number of active sessions created by the node (`numberOfSessionsPerNode`), indexed by IP address (`nodeIpAddress`).

Switch Group

The Switch Group contains objects describing switch ports on the switch. Objects in this group have the object identifier prefix `swi` ({ `modules 87` }).

The following objects are defined:

- `swiPortTable` ({ `swi 1` }) is a table of switch port properties indexed by `swiPortNumber`.
- `swi56xxPortCounterTable` ({ `swi 2` }) is a table of switch port counters for Rapier, AT-8600, AT-8700XL and AT-8800 Series switches, indexed by `swi56xxPortNumber`.
- `swiDebugVariables` ({ `swi 3` }) contains debugging variables.
- `swiPortVlanTable` ({ `swi 4` }) is a table of port/VLAN mappings, indexed by `swiPortVlanPortNumber` and `swiPortVlanVlanId`. It contains the following objects:
 - `swiPortVlanPortNumber`, the index of a port on the switch.
 - `swiPortVlanVlanId`, the VID of the VLAN the port belongs to.
 - `swiPortVlanControl`, the current state of the port in the VLAN. The port can be enabled or disabled in the VLAN by setting `swiPortVlanControl` to enable (1) or disable (2), respectively.
- The `swiPortVlanStateNotify` trap ({ `swi 9` }) is generated when a port in a VLAN changes state, and contains the following objects:
 - `swiPortVlanPortNumber`
 - `swiPortVlanVlanId`
 - `swiPortVlanControl`
- `swiIntrusionDetectionTrap` ({ `swi(87) swiTrap(0) 6` }) is a trap generated when an intrusion is detected on a port, the port has intrusion detection enabled, and the action configured is to generate a trap. It returns the `ifIndex` of the port (`swiPortNumber`).

PIM4 Group

For switches that support PIM, 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.

- pim4Events ({ pim4 0 }) is a collection of traps for the following PIM events:
 - pim4NeighbourAddedTrap ({ pim4Events 1 }) is generated when a PIM neighbour has been added.
 - pim4NeighbourDeletedTrap ({ pim4Events 2 }) is generated when a PIM neighbour has been deleted.
 - pim4InterfaceUpTrap ({ pim4Events 3 }) is generated when a PIM interface has been enabled and is active.
 - pim4InterfaceDownTrap ({ pim4Events 4 }) is generated when a PIM interface has been disabled and is inactive.
 - pim4ErrorTrap ({ pim4Events 5 }) is generated for a PIM error. The error is defined by pim4ErrorTrapType.
- pim4ErrorTrapType ({ pim4 1 }) is the type of the last error that resulted in an error trap being sent.

QoS Group

The QoS group contains a collection of objects for managing QoS. Objects in this group have the object identifier qos ({ modules 99 }).

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.

Load Balancer Group

For switches that support server load balancing, the Load Balancer Group contains objects describing the current configuration and status of load balancing operations on the switch. 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.

DS3 Group

For switches that support DS3, the DS3 Group contains objects for managing DS3. Objects in this group have the object identifier prefix ds3module { modules 109 }.

The following objects are defined:

- ds3TrapTable { ds3module 1 } is a table of DS3 traps.

Port Authentication Group

The Port Authentication group contains additional objects for managing systems using a multi-supplciant configuration that are not included in the Port Access Control (IEEE 802.1X) MIB ([“Port Access Control MIB” on page C-43](#)).

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 C-43](#)), except that:

- The names of the 802.1x objects are prefixed with “atr”.
- There is an additional object to represent the MAC address of a supplicant.
- The names of the MAC based authentication objects are prefixed with “atrMacBased”.
- There are additional objects specific to the Allied Telesis implementation of 802.1x:
 - atrDot1xAuthPreAuthVlan
 - atrDot1xAuthPostAuthVlan
 - atrDot1xAuthLastAuthReason
 - atrDot1XAuthVlanAssignment
 - atrDot1XAuthSecureVlan
 - atrDot1XAuthGuestVlan
 - atrDot1XAuthMibReset
 - atrDot1XAuthTrap
- There are additional trap objects that are sent when an 802.1x or MAC based Authenticator authenticates, unauthenticates, or fails the authentication of a supplicant.

The following objects are defined:

- `atrDot1xPaePortTable` and `atrMacBasedAuthPaePortTable` are tables of system level information (port number, protocol version, and initialisation control) for each port supported by the Port Access Entity.
- `atrDot1xAuthConfigTable` and `atrMacBasedAuthConfigTable` are tables 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 for the Authenticator PAE associated with each Port.
- `atrDot1xAuthenticated` is a trap message generated when an 802.1x Authenticator authenticates a supplicant.
- `atrDot1xUnAuthenticated` is a trap message generated when an 802.1x Authenticator unauthenticates a supplicant.
- `atrDot1xFailedAuth` is a trap message generated when an 802.1x Authenticator fails the authentication of a supplicant.
- `atrMacBasedAuthAuthenticated` is a trap message generated when a MAC based Authenticator authenticates a supplicant.
- `atrMacBasedAuthUnAuthenticated` is a trap message generated when a MAC based Authenticator unauthenticates a supplicant.
- `atrMacBasedAuthFailedAuth` is a trap message generated when a MAC based Authenticator fails the authentication of a supplicant.

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 Telesis 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.

Stack Group

For switches that support stacking, 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 }`) is the ID number of the stack the device belongs to. 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 }`) is 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 }`) is the current status of the stack module on the current SNMP host. The default is disabled (1).

- `stackInterface` ({ stack 4 }) is 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 }) is the authentication type to use with the stack password. 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 }) is the password to authenticate stack messages. By default, this variable holds an empty string. This must be changed before setting the stack authentication method to plaintext or MD5.
- `counters` ({ stack 7 }) is a collection of objects representing the occurrences of specific events to a specific stack host.
- `stackMemberTable` ({ stack 8 }) is a table of information 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.

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 switch.
- `sqosTrafficTable` ({ sqos 3 }) is a table of information about each traffic class that is configured on the switch, 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 switch. 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 switch, 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 switch. 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 switch. 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 switch.

DAR objects recognise sessions such as VoIP phone calls and real-time video and create dynamic classifiers to match traffic belonging to these sessions.

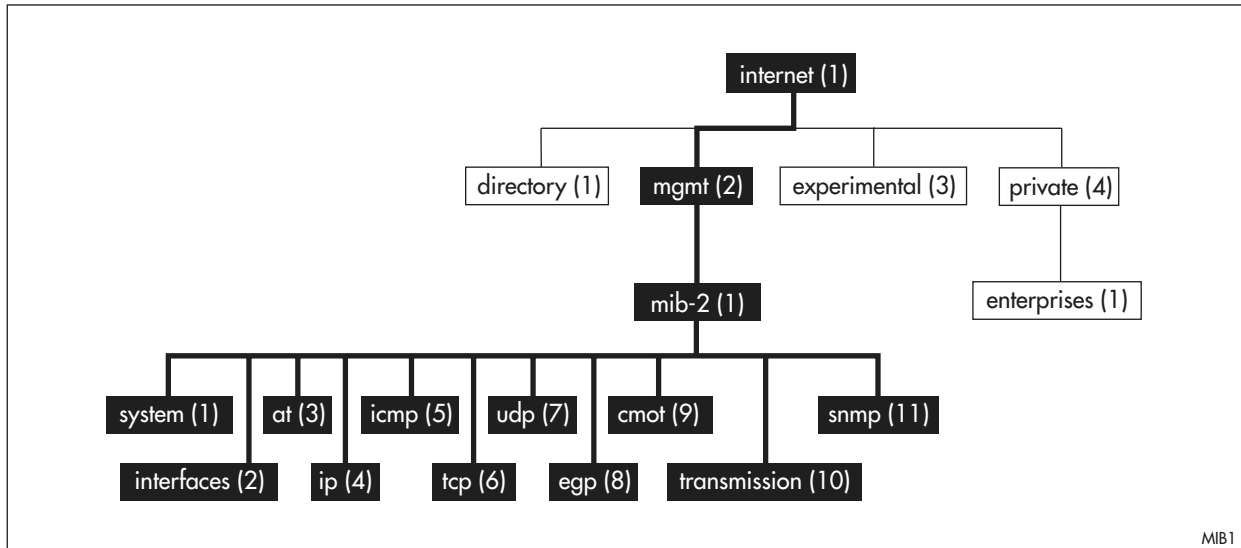
- `sqosInterfaceTable ({ sqos 9 })` is a table that lists the policies and DAR objects attached to interfaces on the switch.
- `sqosPolicyCounterTable ({ sqos 10 })` is a table that displays policy counters as packets and bytes, and the interface and traffic direction the policy applies to.
- `sqosTcCounterTable ({ 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) subtree (Figure C-3) and have the object identifier prefix mib-2 ({ mgmt 1 }).

Figure C-3: The MIB-II subtree 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 switch models implement all groups in MIB-II. However, the implementation of some objects differs from RFC 1213 ([Table C-7](#)). In particular, some read-write objects are implemented as read-only.

Table C-7: MIB-II implementation variations

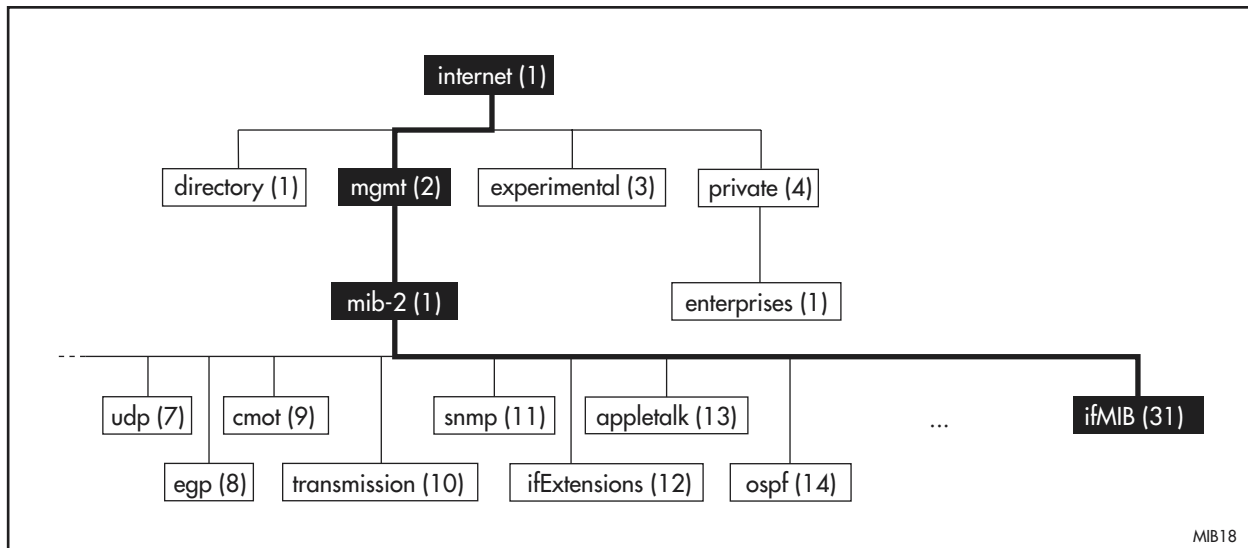
Object Name	Object ID	Access	Implementation
ifAdminStatus	{ 1.3.6.1.2.1.2.2.1.7 }	Read-write	Read-write for PPP, VLAN and Synchronous interfaces Read-only for all other interfaces
atfIndex	{ 1.3.6.1.2.1.3.1.1.1 }	Read-write	Read-only
atPhysAddress	{ 1.3.6.1.2.1.3.1.1.2 }	Read-write	Read-only
atNetAddress	{ 1.3.6.1.2.1.3.1.1.3 }	Read-write	Read-only
ipRouteDest	{ 1.3.6.1.2.1.4.21.1.1 }	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

Interfaces Group MIB

RFC 2863, *The Interfaces Group MIB*, defines a portion of the Management Information Base (MIB) for managing network interfaces.

Objects defined in this MIB reside in the mib(1) subtree (Figure C-4) and have the object identifier prefix ifMIB ({ mib-2 31 }).

Figure C-4: The Interface Extensions subtree of the Internet-standard Management Information Base (MIB)



The Interface Extensions MIB consists of the following objects:

- ifNumber is the number of network interfaces present on the system, and redefines (replaces) the ifNumber object in MIB-II.
- ifTableLastChange is the value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable.
- The Interface Table (ifTable) is a table of information about network interfaces present on the system, and redefines (replaces) the ifTable object in MIB-II.
- The Interface Extensions Table (ifXTable) is a table of additional information about the network interfaces present on the system, and redefines (replaces) the ifExtnsTable object in MIB-II. It includes 64-bit versions of the basic ifTable counters for high capacity interfaces.
- The Interface Stack Table (ifStackTable) is a table of information about the sub-layers of network interfaces present on the system, and in particular, which sub-layers run on top of other sub-layers.
- The Interface Test Table (ifTestTable) is a table of objects for configuring and monitoring tests on interfaces present on the system.
- The Generic Receive Address Table (ifRcvAddressTable) is a table of information about the broadcast, multicast and unicast addresses that the system will receive packets/frames for on a particular interface. It redefines (replaces) the ifExtnsRcvAddrTable object in MIB-II.

Implementation

All switch models implement the following objects in the Interfaces Group MIB:

- ifNumber
- The Interface Table (ifTable)
- The Interface Extensions Table (ifXTable)
- The Interface Stack Table (ifStackTable)
- The Generic Receive Address Table (ifRcvAddressTable)

However, the implementation of some objects differs from RFC 2863 (Table C-8).

The following objects in the Interface Extensions MIB are not implemented:

- The Interface Test Table (ifTestTable)

Table C-8: Interface Extensions MIB implementation variations

Object Name	Object ID	Access	Implementation
ifPromiscuousMode	{ 1.3.6.1.2.1.31.1.1.16 }	Read-write	Read-only
ifStackStatus	{ 1.3.6.1.2.1.31.1.2.1.3 }	Read-write	Read-only
ifRcvAddressAddress	{ 1.3.6.1.2.1.31.1.4.1.1 }	Read-write	Read-only
ifRcvAddressStatus	{ 1.3.6.1.2.1.31.1.4.1.2 }	Read-write	Read-only
ifRcvAddressType	{ 1.3.6.1.2.1.31.1.4.1.3 }	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) subtree, under the IP Group defined in MIB-II ([Figure C-3 on page C-29](#)) 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

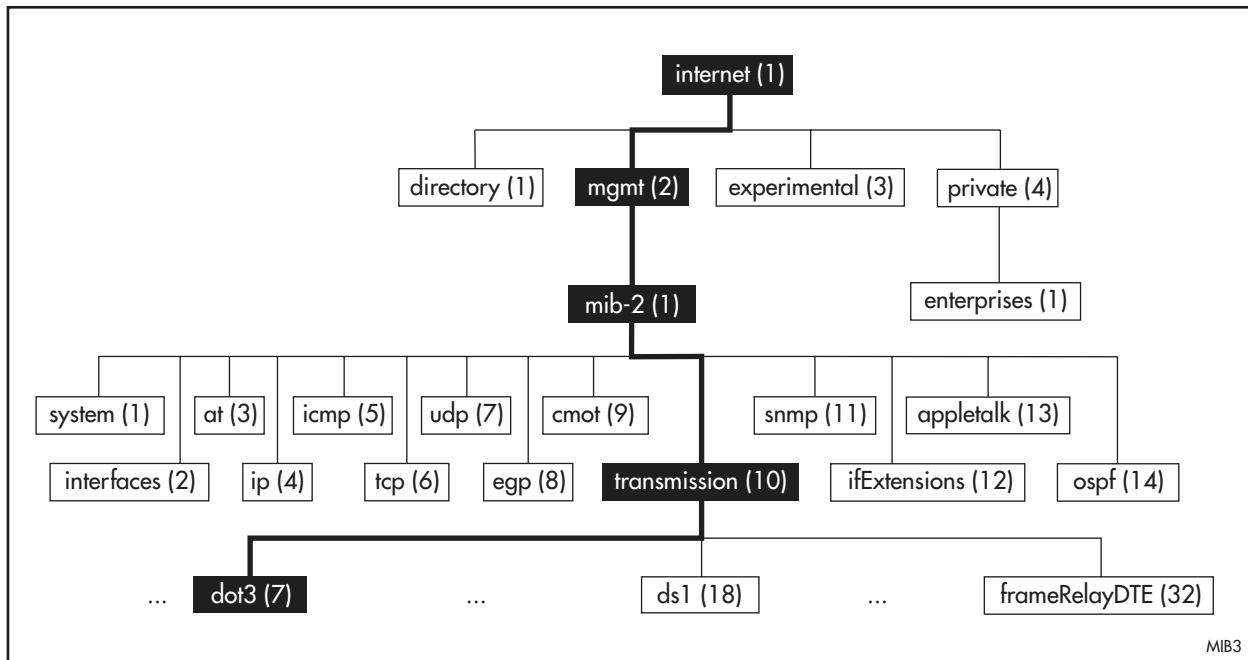
All switch models implement all objects in the IP Forwarding Table MIB.

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) subtree, under the Transmission Group defined in MIB-II (Figure C-5) and have the object identifier prefix dot3 ({ transmission 7 }).

Figure C-5: The Ethernet-like interface types subtree 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

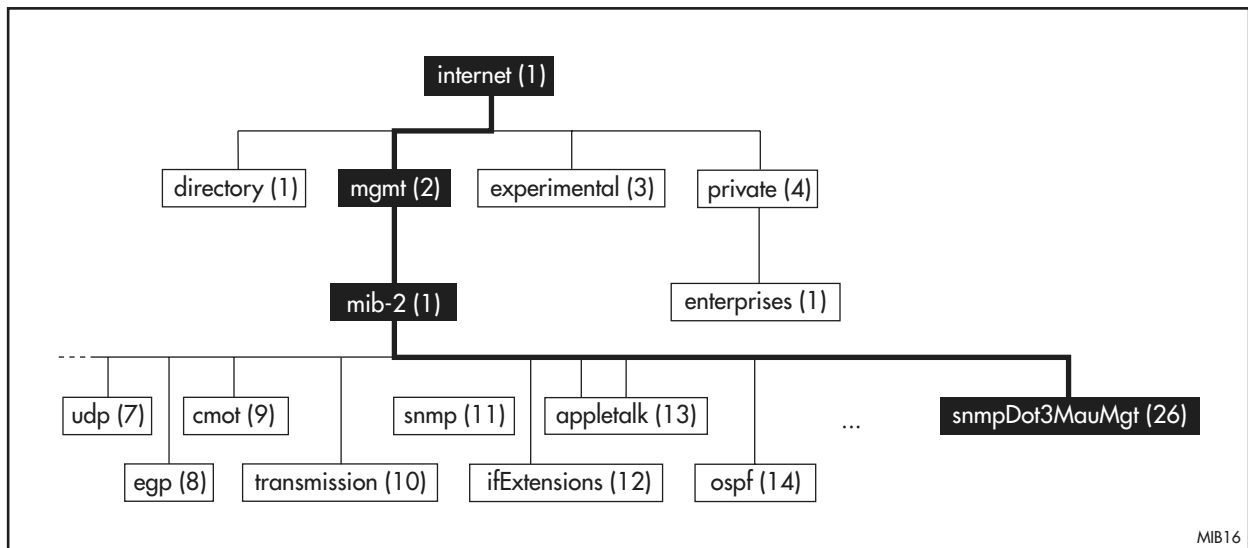
All switch models implement all objects in the Ethernet-like Interface Types MIB.

Medium Attachment Unit (MAU) MIB

RFC 2239, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) using SMIv2*, defines a portion of the Management Information Base (MIB) for managing 10Mbps and 100Mbps Medium Attachment Units (MAUs) based on IEEE Std 802.3.

Objects defined in this MIB reside in the mib(1) subtree (Figure C-6) and have the object identifier prefix snmpDot3MauMgt ({ mib-2 26 }).

Figure C-6: The Medium Attachment Unit (MAU) subtree of the Internet-standard Management Information Base (MIB)



The Medium Attachment Unit MIB consists of the following objects:

- dot3RpMauBasicGroup is a collection of objects for managing IEEE 802.3 repeaters:
 - rpMauTable is a table of information about the MAU(s) attached to the ports of a repeater.
 - rpJackTable is a table of information about the external jacks attached to MAUs attached to the ports of a repeater, and applies to MAUs attached to repeaters which have one or more external jacks (connectors).
- dot3IfMauBasicGroup is a collection of objects for managing IEEE 802.3 Ethernet-like interfaces:
 - ifMauTable is a table of information about MAU(s) attached to an interface. The value of ifMauIfIndex is the same as the value of ifIndex in the Interface group of MIB-II.
 - ifJackTable is a table of information about the external jacks attached to MAUs attached to an interface, and applies to MAUs attached to interfaces which have one or more external jacks (connectors).
- dot3BroadMauBasicGroup is a collection of objects for managing broadband MAUs:
 - broadMauBasicTable is a table of information about the broadband MAUs connected to interfaces.
- dot3MauType is a collection of object identities for MAU types.

- dot3IfMauAutoNegGroup is a collection of objects for managing autonegotiation:
 - ifMauAutoNegTable is a table of information about the autonegotiation function of MAUs attached to interfaces.

Implementation

All switch models implement all objects in the following groups in the Medium Attachment Unit MIB:

- The Interface MAU Basic Group (dot3IfMauBasicGroup)
- The dot3MauType object identities
- The Interface MAU Autonegotiation Group (dot3IfMauAutoNegGroup)

However, the implementation of some objects differs from RFC 2239 ([Table C-9](#)).

The following groups in the Medium Attachment Unit MIB are not implemented because they are not meaningful in the context of the switch:

- The Repeater MAU Basic Group (dot3RpMauBasicGroup)
- The Broadband MAU Basic Group (dot3BroadMauBasicGroup)

Table C-9: Medium Attachment Unit (MAU) MIB implementation variations

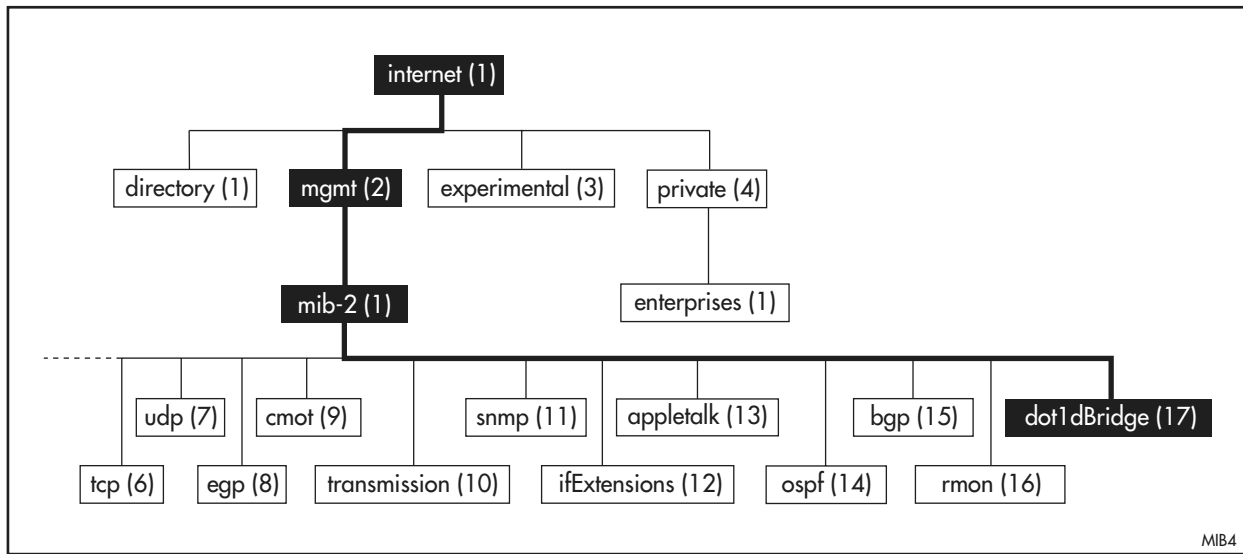
Object Name	Object ID	Access	Implementation
ifMauStatus	{ 1.3.6.1.2.1.26.2.1.1.4 }	Read-write	Read-only
ifMauDefaultType	{ 1.3.6.1.2.1.26.2.1.1.11 }	Read-write	Read-only
ifMauAutoNegAdminStatus	{ 1.3.6.1.2.1.26.5.1.1.1 }	Read-write	Read-only
ifMauAutoNegCapAdvertised	{ 1.3.6.1.2.1.26.5.1.1.6 }	Read-write	Read-only
ifMauAutoNegRestart	{ 1.3.6.1.2.1.26.5.1.1.8 }	Read-write	Read-only

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) subtree (Figure C-7) and have the object identifier prefix dot1dBridge ({ mib-2 17 }).

Figure C-7: The bridge subtree 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

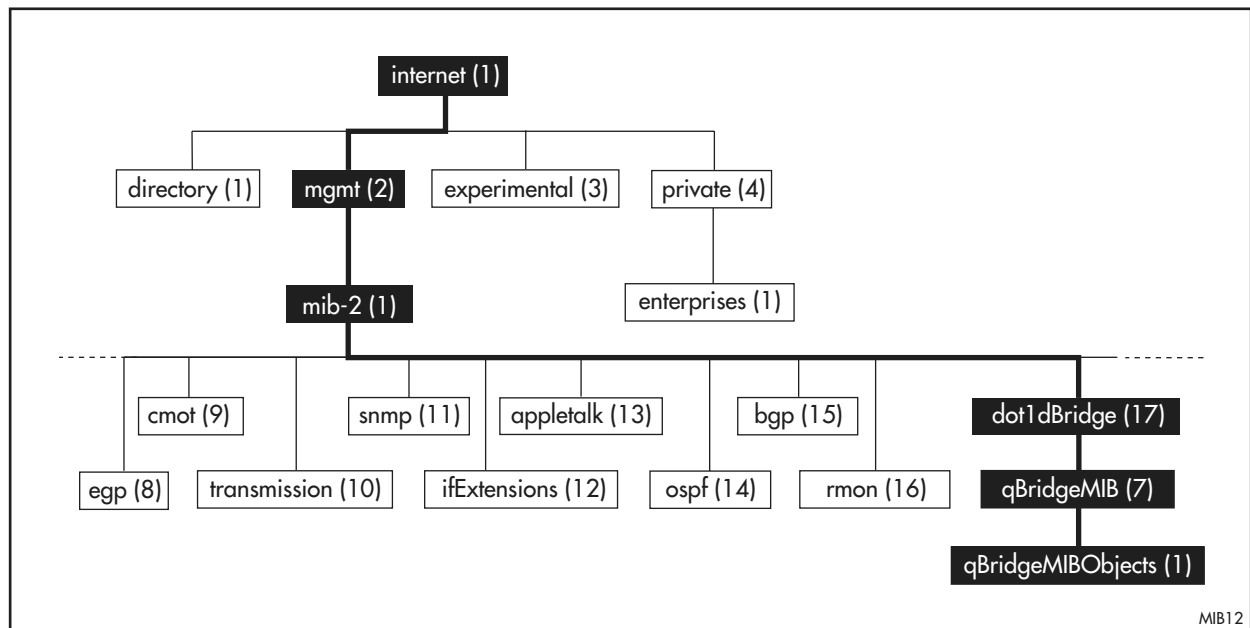
All switch models implement all groups in the Bridge MIB.

Virtual Bridge (VLAN) MIB

RFC 2674, *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*, defines a portion of the Management Information Base (MIB) for managing IEEE Standard 802.1Q VLANs.

Objects defined in this MIB reside in the mib(1) subtree (Figure C-8), under the dot1dBridge subtree defined in RFC 1493 (see “Bridge MIB” on page C-37), and have the object identifier qBridgeMIBObjects ({ mib-2 dot1dBridge(17) qBridgeMIB(7) 1 }).

Figure C-8: The Virtual Bridge (VLAN) subtree of the Internet-standard Management Information Base (MIB)



The MIB is organised into four logical groups:

- The dot1qBase Group contains general objects that apply to any device that supports IEEE Standard 802.1Q VLANs.
- The dot1qTp Group contains objects that describe the operation and status of transparent bridging, including the dynamic filtering databases for unicast and multicast forwarding.
- The dot1qStatic Group contains objects that describe static configurations for transparent bridging, including static entries in the filtering databases for unicast and multicast forwarding.
- The dot1qVlan Group contains objects that describe the configuration and status of VLANs, including statically configured VLANs and VLANs configured dynamically by protocols like GVRP.

Implementation

The switch implements the following objects and groups in the MIB:

- All objects in the dot1qBase Group.
- The dot1qFdbTable object in the dot1qTp Group.
- The dot1qTpFdbTable object in the dot1qTp Group.
- The dot1qVlanNumDeletes object in the dot1qVlan Group.

- The dot1qVlanCurrentTable object in the dot1qVlan Group.
- The dot1qVlanStaticTable object in the dot1qVlan Group.
- The dot1qNextFreeLocalVlanIndex object in the dot1qVlan Group.
- The dot1qPortVlanTable object in the dot1qVlan Group.

The following objects and groups in the MIB are not implemented:

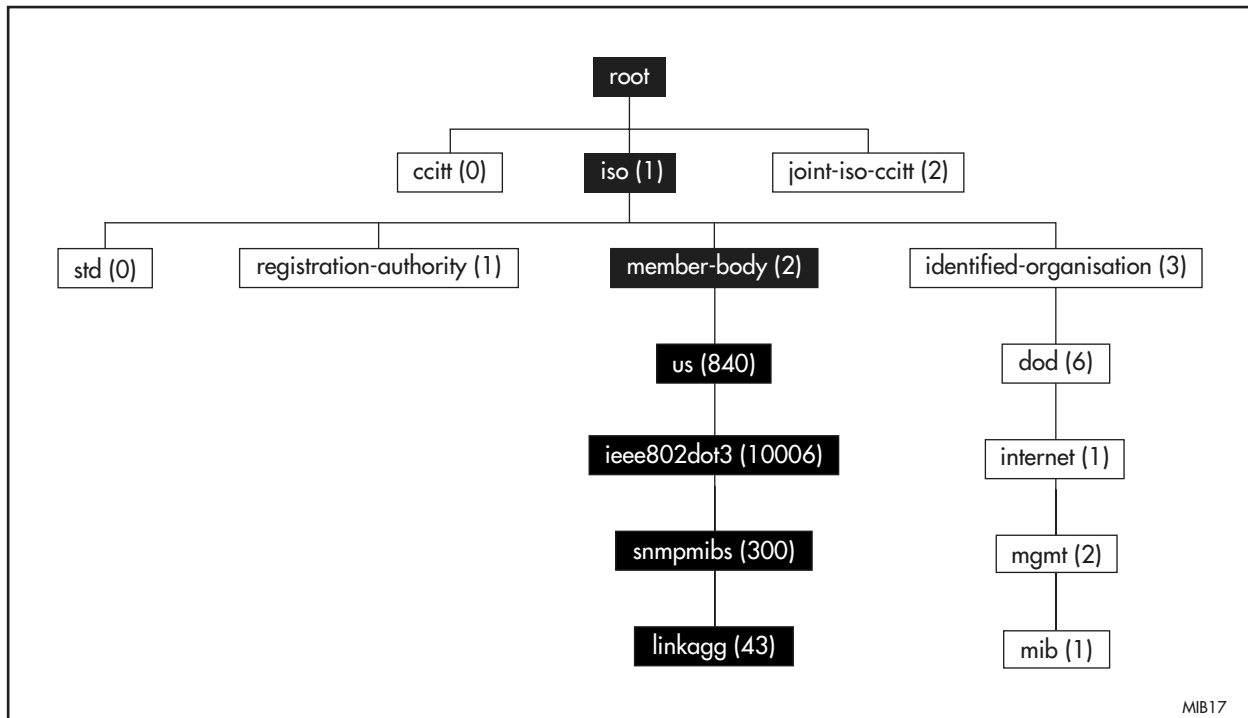
- The dot1qTpGroupTable object in the dot1qTp Group.
- The dot1qForwardAllTable object in the dot1qTp Group.
- The dot1qForwardUnregisteredTable object in the dot1qTp Group.
- All objects in the dot1qStatic Group.
- The dot1qPortVlanStatisticsTable object in the dot1qVlan Group.
- The dot1qPortVlanHCStatisticsTable object in the dot1qVlan Group.
- The dot1qLearningConstraintsTable object in the dot1qVlan Group.
- The dot1qConstraintSetDefault object in the dot1qVlan Group.
- The dot1qConstraintTypeDefault object in the dot1qVlan Group.
- The qBridgeConformance object in the dot1qVlan Group.

Link Aggregation MIB

IEEE Standard 802.3-2002 Part 3, *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, defines a portion of the Management Information Base (MIB) for managing IEEE Standard 802.3 link aggregation.

Objects defined in this MIB reside in the { iso(1) member-body(2) us(840) ieee802dot3(10006) snmpmibs(300) } subtree (Figure C-17) and have the object identifier linkagg ({ snmpmibs 43 }).

Figure C-9: The Link Aggregation subtree of the Internet-standard Management Information Base (MIB)



The MIB is organised into four logical groups:

- The Aggregator Group contains objects for configuring and monitoring an aggregator.
- The Aggregation Port Group contains objects for configuring and monitoring a port for link aggregation.
- The Aggregation Port Statistics Group contains objects for collecting statistical information about LACP and Marker protocol activity on a port.
- The Aggregation Port Debug Group contains objects for debugging LACP and detecting fault conditions on a port.

Each aggregator is uniquely identified by an interface number, dot3adAggIndex, which has the same value as the ifIndex of the corresponding entry in the ifTable of the Interfaces group in MIB-II.

Each aggregation port is uniquely identified by an interface number, dot3adAggPortIndex, which has the same value as the ifIndex of the corresponding entry in the ifTable of the Interfaces group in MIB-II.

Implementation

All switch models implement all objects in the Link Aggregation MIB. However, the implementation of some objects differs from the IEEE standard (Table C-10). The marker protocol is not supported so all associated statistics will return zero (0).

Table C-10: Link Aggregation MIB implementation variations

Object Name	Object ID	Access	Implementation
dot3adAggCollectorMaxDelay	{ 1.2.840.10006.300.43.1.1.1.11 }	Read-write	Read-only
dot3adAggPortActorOperKey	{ 1.2.840.10006.300.43.1.2.1.1.5 }	Read-write	Read-only
dot3adAggPortPartnerAdminSystemPriority	{ 1.2.840.10006.300.43.1.2.1.1.6 }	Read-write	Read-only
dot3adAggPortPartnerAdminSystemID	{ 1.2.840.10006.300.43.1.2.1.1.8 }	Read-write	Read-only
dot3adAggPortPartnerAdminKey	{ 1.2.840.10006.300.43.1.2.1.1.10 }	Read-write	Read-only
dot3adAggPortPartnerAdminPort	{ 1.2.840.10006.300.43.1.2.1.1.16 }	Read-write	Read-only
dot3adAggPortPartnerAdminPortPriority	{ 1.2.840.10006.300.43.1.2.1.1.18 }	Read-write	Read-only
dot3adAggPortActorAdminState	{ 1.2.840.10006.300.43.1.2.1.1.20 }	Read-write	Read-only
dot3adAggPortPartnerAdminState	{ 1.2.840.10006.300.43.1.2.1.1.22 }	Read-write	Read-only
dot3adAggPortStatsMarkerPDUsRx	{ 1.2.840.10006.300.43.1.2.1.1.2 }	Read-only	Always returns 0
dot3adAggPortStatsMarkerResponsePDUsRx	{ 1.2.840.10006.300.43.1.2.1.1.3 }	Read-only	Always returns 0
dot3adAggPortStatsMarkerPDUsTx	{ 1.2.840.10006.300.43.1.2.1.1.7 }	Read-only	Always returns 0
dot3adAggPortStatsMarkerResponsePDUsTx	{ 1.2.840.10006.300.43.1.2.1.1.8 }	Read-only	Always returns 0

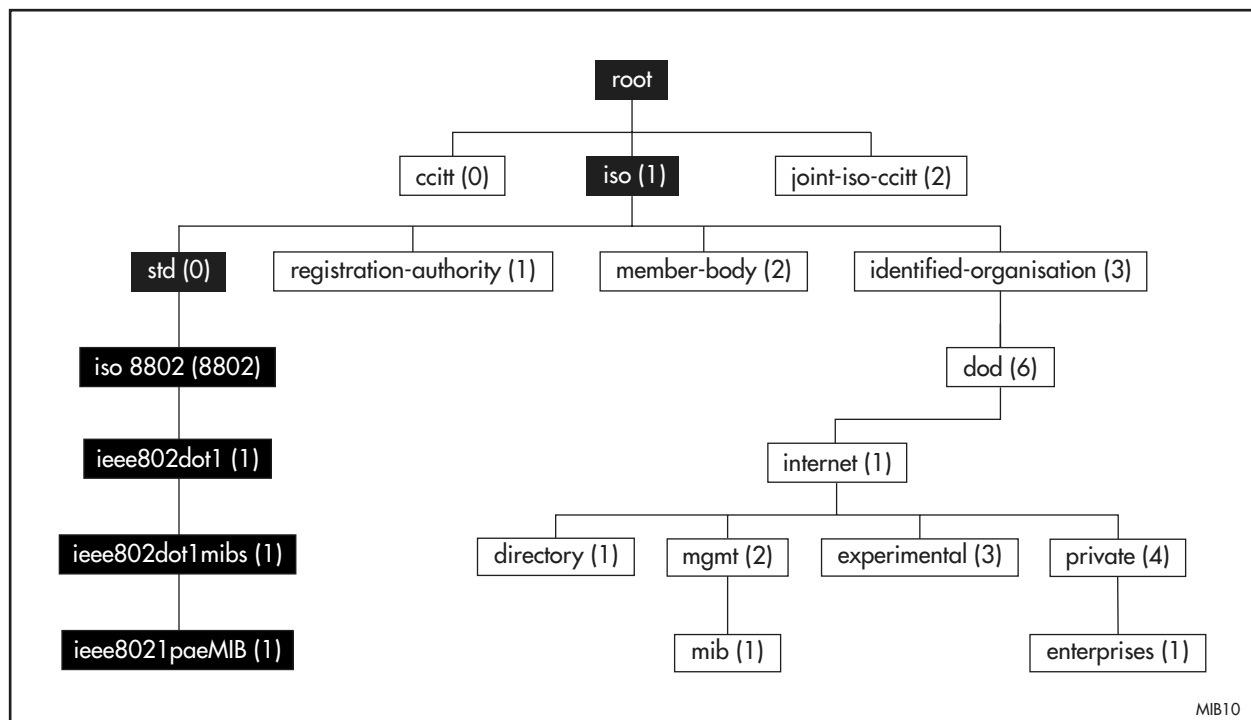
Port Access Control MIB

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 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) ieee802dot1(1) ieee802dot1mibs(1) } subtree (Figure C-10) and have the object identifier prefix ieee8021paeMIB ({ ieee802dot1mibs 1 }).

Figure C-10: 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 switch implements:

- dot1xPaePortTable

In the PAE Authenticator Group, the switch implements:

- dot1xAuthConfigTable
- dot1xAuthStatsTable

and does not implement the following tables:

- dot1xAuthDiagTable
- dot1xAuthSessionStatsTable

In the PAE Supplicant Group, the switch implements:

- dot1xSuppConfigTable
- dot1xSuppStatsTable

All tables in this MIB are indexed by port number.

Table C-11: Port Access Control MIB implementation variations

Object Name	Object ID	Access	Implementation
dot1xAuthAdminControlledDirections	{ 1.0.8802.1.1.1.2.1.1.3 }	Read-write	The current value of the administrative controlled directions parameter for the Port. Read-only.
dot1xAuthKeyTxEnabled	{ 1.0.8802.1.1.1.2.1.1.14 }	Read-write	The value of the keyTransmissionEnabled constant currently in use by the Authenticator PAE state machine. Read-only.

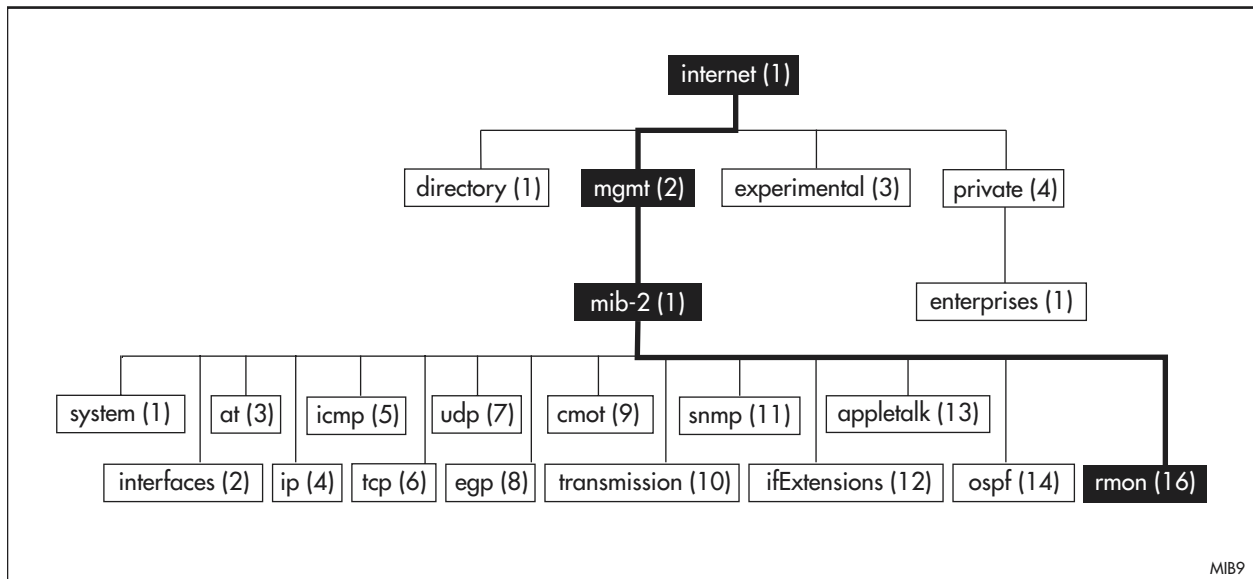
The switch also implements a Port Authentication MIB in the Allied Telesis Enterprise MIB for systems that use a multi-supplicant configuration (see [“Port Authentication Group”](#) on page C-25).

RMON MIB

RFC 2819, *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) subtree (Figure C-11), 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 C-11: RMON in the subtree 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 switch implements the Statistics, History, Alarm, and Event groups of the RMON MIB.

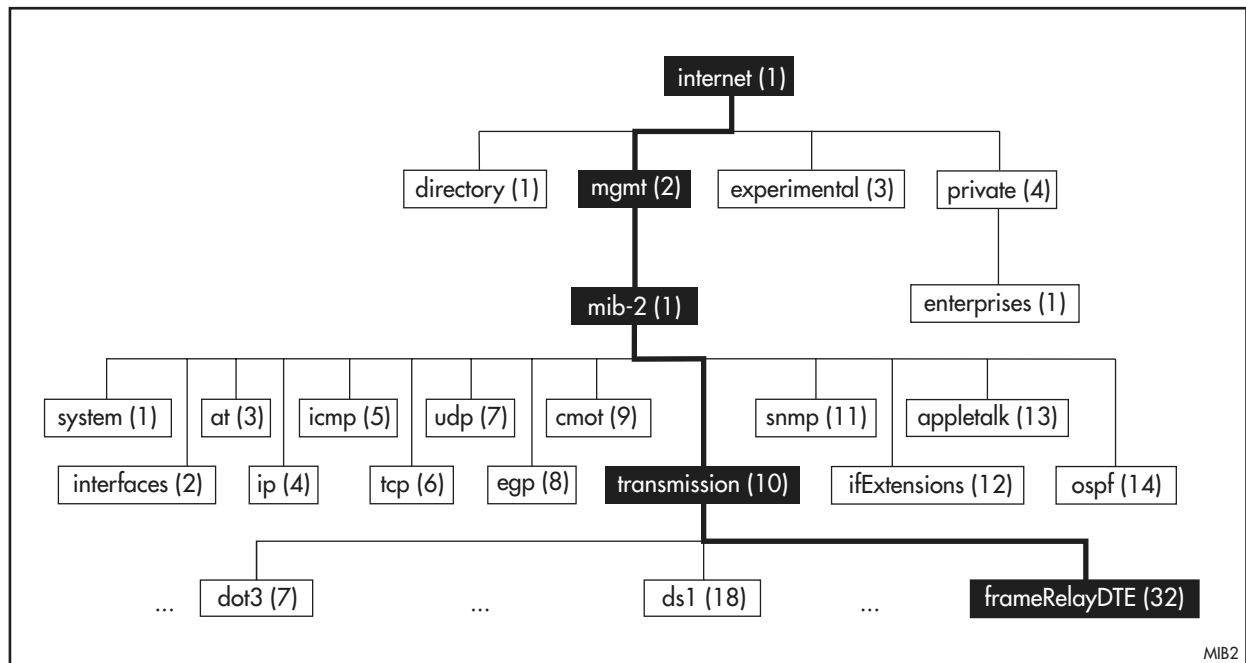
The RMON MIB gathers statistics while SNMP is enabled ([Chapter 55, Simple Network Management Protocol \(SNMP\)](#)).

Frame Relay DTE MIB

RFC 2115, *Management Information Base for Frame Relay DTEs Using SMIV2*, defines a portion of the Management Information Base (MIB) for managing Frame Relay.

Objects defined in this MIB reside in the mib(1) subtree (Figure C-12), under the Transmission Group defined in MIB-II and have the object identifier prefix frameRelayDTE (1 transmission 32).

Figure C-12: The Frame Relay DTE subtree 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 switch 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 2115 ([Table C-12](#)). In particular, some read-write objects are implemented as read-only and traps are not implemented.

Table C-12: Frame Relay DTE MIB implementation variations

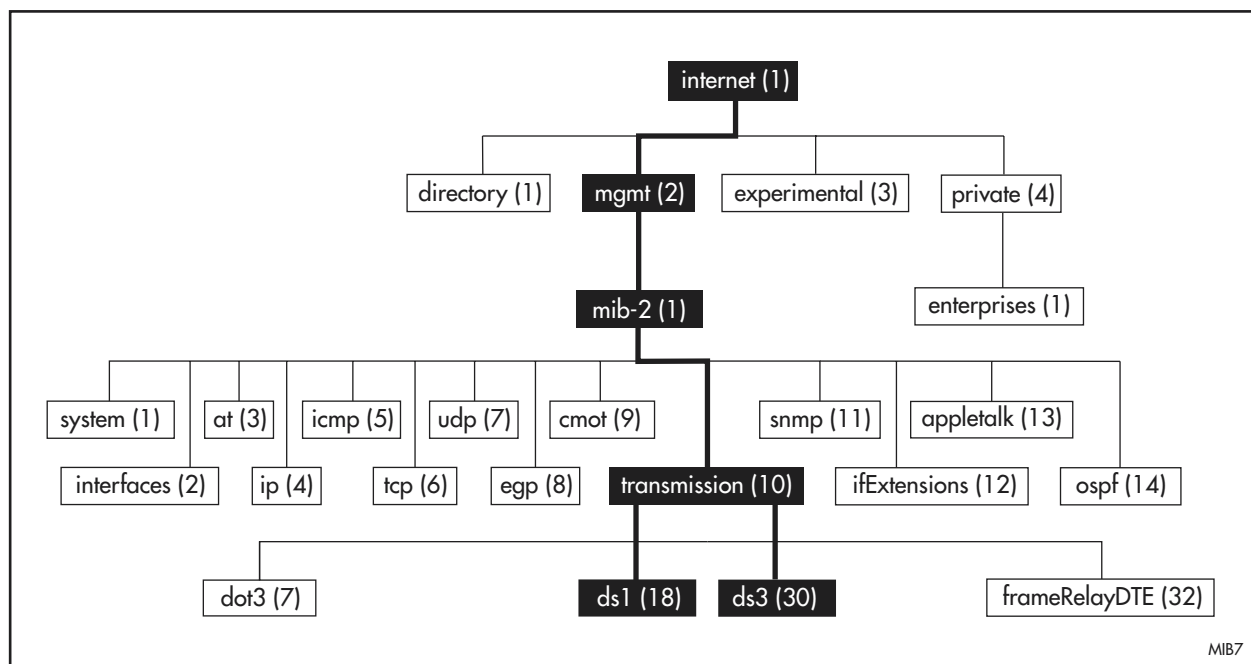
Object Name	Object ID	Access	Implementation
frDlcmiState	{ 1.3.6.1.2.1.10.32.1.1.2 }	Read-write	Read-only
frDlcmiAddress	{ 1.3.6.1.2.1.10.32.1.1.3 }	Read-write	Read-only
frDlcmiAddressLen	{ 1.3.6.1.2.1.10.32.1.1.4 }	Read-write	Read-only
frDlcmiPollingInterval	{ 1.3.6.1.2.1.10.32.1.1.5 }	Read-write	Read-only
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) subtree (Figure C-13), 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).

Figure C-13: The DS1, E1, DS2, E2, DS3 and E3 Interface Types subtrees of the Internet-standard Management Information Base (MIB)



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 switch 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.

DS3/E3 Interface Types MIB

RFC 2496, *Definitions of Managed Objects for the DS3/E3 Interface Type*, defines a portion of the Management Information Base (MIB) for managing DS3 and E3 interfaces.

Objects defined in this MIB reside in the mib(1) subtree ([Figure C-13 on page C-49](#)), under the Transmission Group defined in MIB-II and have the object identifier ds3 ({ transmission 30 }). These objects are used when the particular media being used to realise an interface is a DS3/E3 interface. At present, this applies to ifType objects with the value ds3 (30).

The DS3 definitions in this MIB are based on the DS3 specifications in ANSI T1.102-1987, ANSI T1.107-1988, ANSI T1.107a-1990, and ANSI T1.404-1989 [8,9,9a,10]. The E3 definitions are not applicable as only ds3 is supported.

The MIB is organised into five logical groups:

- The DS3 Near End Group contains objects that describe the configuration and performance statistics of the near end of the DS3 link.
- The DS3 Far End Group contains objects that describe the configuration and performance statistics of the far end of the DS3 link.
- The DS3 Fractional Table contains objects that describe which DS3 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 DS3 Trap Group contains a trap for line status changes.
- The Conformance Group contains objects that describe compliance statements and mandatory object groups.

Implementation

Rapier 24*i* models implement the following objects and groups in the DS3/E3 Interface Types MIB:

- All objects in the DS3 Near End Group, except dsx3IfIndex in the dsx3ConfigTable because this object is equal to dsx3LineIndex
- All objects in the DS3 Far End Group.
- All traps in the DS3 Trap Group.

The following objects and groups in the DS3/E3 Interface Types MIB are not implemented:

- All objects in the DS3 Fractional Table.
- All objects in the Conformance Group.

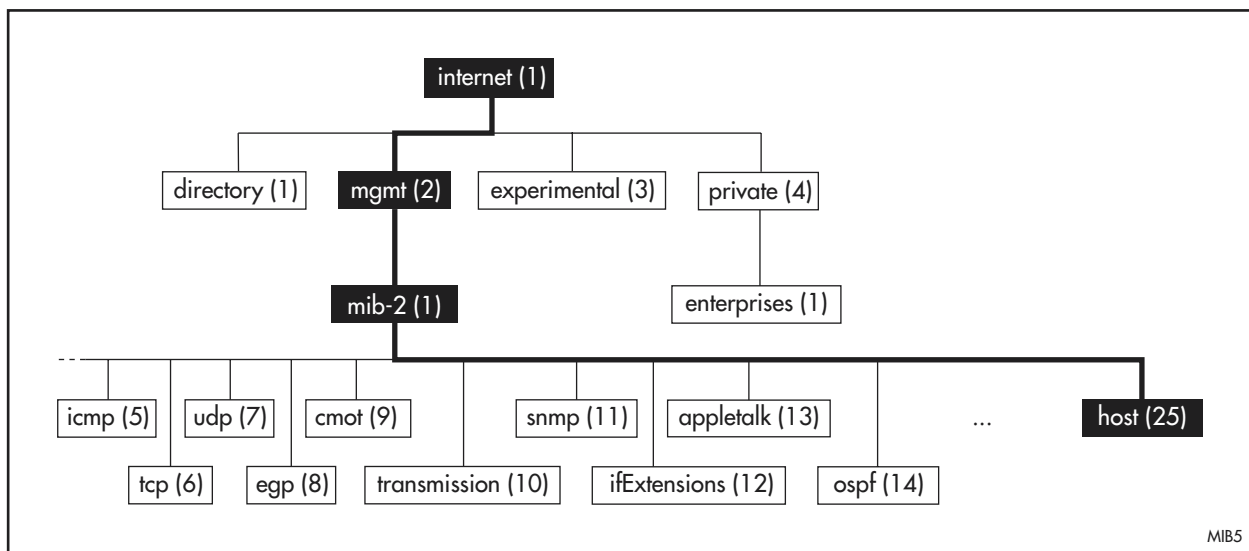
Host Resources MIB

RFC 2790, *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) subtree (Figure C-14) and have the object identifier prefix host ({ mib-2 25 }).

Figure C-14: The Host Resources subtree 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 switch 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 2790 ([Table C-13](#)).

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

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

Table C-13: 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 1
hrSystemMaxProcesses	{ 1.3.6.1.2.1.25.1.7 }	Read-only	Always returns 1
hrStorageAllocationUnits	{ 1.3.6.1.2.1.25.2.3.1.4 }	Read-only	Returns 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 C-14 on page C-54 lists the devices supported by the switch
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 C-9
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 switch provides print services 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 switch 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 switch does not support any file systems close enough in semantics to those catered for by this MIB.

Table C-14: Host Resources MIB device types supported by the switch

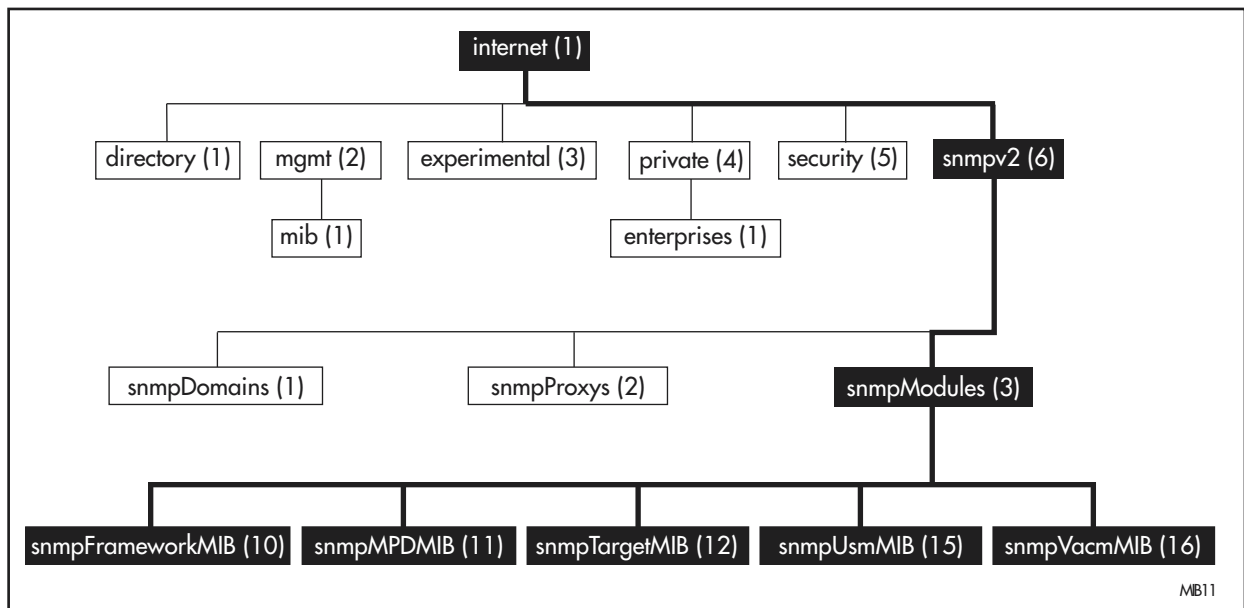
hrDeviceType	Switch Device
hrDeviceProcessor	Processor
hrDeviceNetwork	LAN/WAN network port
hrDeviceCoprocesor	MAC card
hrDeviceSerialPort	Asynchronous ports
hrDeviceClock	RTC
hrDeviceVolatileMemory	RAM, FSRAM, CAM
hrDeviceNonVolatileMemory	NVS, flash

SNMP v3 MIBs

The SNMPv3 subtrees define a portion of the Management Information Base (MIB) for managing the Simple Network Management Protocol version 3.

Objects in these subtrees reside in the { iso(1) identified-organisation(3) dod(6) internet(1) snmpv2(6) } subtree (Figure C-15) and have the object identifier prefix snmpModules ({ snmpv2 3 }).

Figure C-15: SNMPv3 subnet-tree of the Internet-standard Management Information Base (MIB)



Objects in the SNMPv3 subtrees are defined by the following MIBs:

- The SNMP-FRAMEWORK-MIB comprises the snmp Engine, which contains information about managed objects used to instrument SNMP entities. This MIB is defined RFC 3411.
- The 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.
- The Management Target MIB provides 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.
- The User-based Security MIB 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.
- The View-based Access Control MIB 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

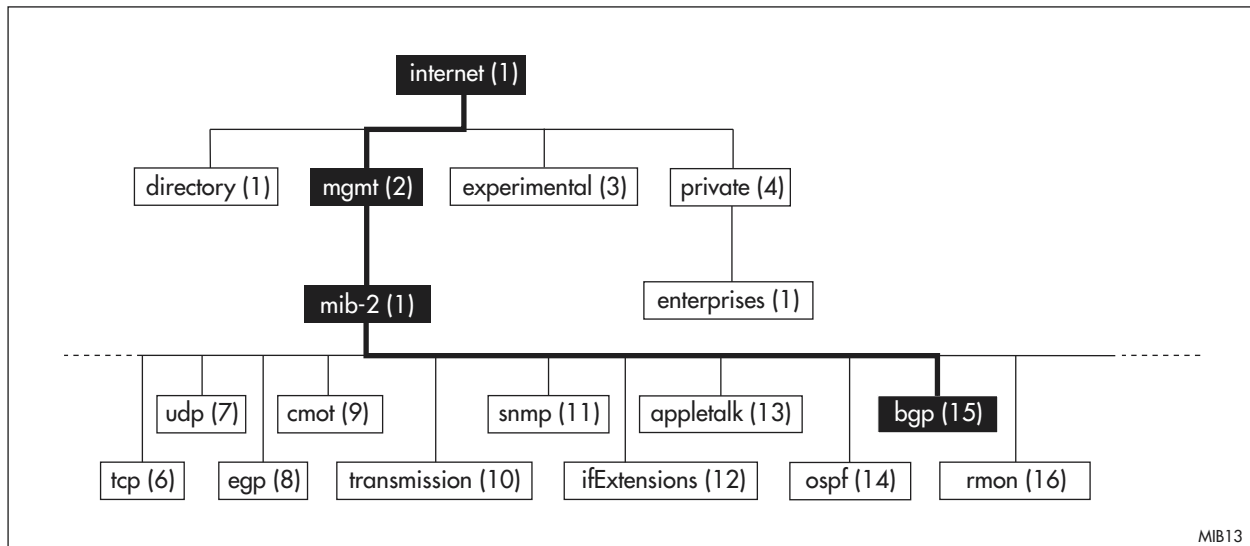
All switch models implement all objects in the SNMPv3 MIBs.

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) subtree (Figure C-16) and have the object identifier prefix bgp ({ mib-2 15 }).

Figure C-16: The BGP-4 subtree 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

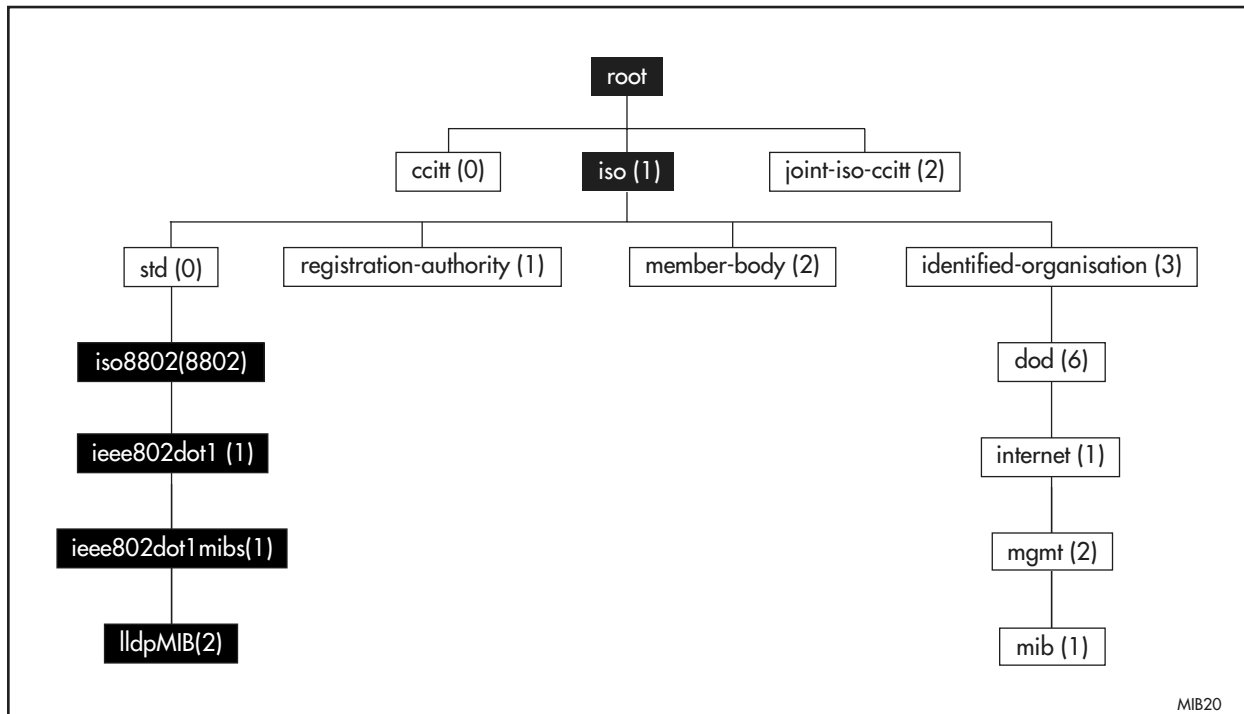
All switch models implement all objects except the BGP Received Path Attribute Table (bgpRcvdPathAttrTable) which has been deprecated and does not apply to BGP version 4.

Link Layer Discovery Protocol (LLDP) MIB

IEEE Standard 802.1AB-2005, *IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery*, defines a portion of the Management Information Base (MIB) for managing IEEE Standard 802.1AB link layer discovery.

Objects defined in this MIB reside in the { iso(1) std(0) iso8802(8802) ieee802dot1(1) ieee802dot1mibs(1) } subtree (Figure C-17) and have the object identifier lldpMib ({ ieee802dot1mibs 2 }).

Figure C-17: The Link Layer Discovery subtree of the Internet-standard Management Information Base (MIB)



The MIB is organised into the following groups:

- lldpConfiguration { lldpMIB(2) lldpObjects(1) 1 } contains objects for configuring LLDP.
- lldpStatistics { lldpMIB(2) lldpObjects(1) 2 } contains objects for collecting statistics about the operation of LLDP.
- lldpLocalSystemData { lldpMIB(2) lldpObjects(1) 3 } contains objects that describe the local system.
- lldpRemoteSystemsData { lldpMIB(2) lldpObjects(1) 4 } contains objects that describe remote systems.
- lldpNotifications { lldpMIB(2) 0 } contains traps for monitoring changes to remote systems.

Implementation

All switch models implement all objects in the Link Layer Discovery Protocol MIB.

State and Topology Change Traps

This section describes additional traps for protocol state or network topology changes supported by the switch.

OSPF RFC 1850, *OSPF Version 2 Management Information Base*, defines a portion of the Management Information Base (MIB) for managing Version 2 of the Open Shortest Path First Routing Protocol.

Objects defined in this MIB reside in the mib(1) subtree (Figure C-3) and have the object identifier prefix ospf ({ mib-2 14 }).

The switch implements the following traps from the ospfTrap(16) ospfTraps(2) subtree of the OSPF Version 2 MIB:

- The ospfIfStateChange trap ({ ospfTraps 16 }) is generated when a non-virtual OSPF interface changes state, and contains the following objects:
 - ospfRouterId, the router ID of the originator of the trap
 - ospfIfIpAddress, the IP address of the interface that changed state, for interfaces with an IP address
 - ospfAddressLessIf, the ifIndex of the interface that changed state, for addressless interfaces
 - ospfIfState, the new state of the interface
- The ospfVirtIfStateChange trap ({ ospfTraps 1 }) is generated when a virtual OSPF interface changes state, and contains the following objects:
 - ospfRouterId, the router ID of the originator of the trap
 - ospfVirtIfAreaId, the transit area used by the virtual interface
 - ospfVirtIfNeighbor, the router ID of the virtual neighbour
 - ospfVirtIfState, the new state of the virtual interface
- The ospfNbrStateChange trap ({ ospfTraps 2 }) is generated when a non-virtual OSPF neighbour changes state, and contains the following objects:
 - ospfRouterId, the router ID of the originator of the trap
 - ospfNbrIpAddress, the IP address the neighbour uses as its IP source address
 - ospfNbrAddressLessIndex, the ifIndex of the interface the neighbour is attached to, for addressless interfaces
 - ospfNbrRtrId, the router ID of the neighbour
 - ospfNbrState, the new state of the neighbour
- The ospfVirtNbrStateChange trap ({ ospfTraps 3 }) is generated when a virtual OSPF neighbour changes state, and contains the following objects:
 - ospfRouterId, the router ID of the originator of the trap
 - ospfVirtNbrArea, the transit area identifier
 - ospfVirtNbrRtrId, the router ID of the virtual neighbour
 - ospfVirtNbrState, the new state of the virtual neighbour

VRRP RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*, defines a portion of the Management Information Base (MIB) for managing the Virtual Router Redundancy Protocol (VRRP).

Objects defined in this MIB reside in the mib(1) subtree ([Figure C-3](#)) and have the object identifier prefix vrrpMIB ({ mib-2 68 }).

The switch implements the following trap from the vrrpNotifications(0) subtree of the VRRP MIB:

- The vrrpTrapNewMaster trap ({ vrrpNotifications 1 }) is generated when the sending agent becomes the new VRRP master, and contains the following object:
 - vrrpOperMasterIpAddr, the primary IP address of the new master

MSTP The IEEE draft ruzin-mstp-mib-04, defines a portion of the Management Information Base (MIB) for managing Multiple and Rapid Spanning Tree Protocols.

Objects defined in this MIB reside in the dot1dBridge subtree defined in RFC 1493 (see [“Bridge MIB” on page C-37](#)), and have the object identifier mstp ({ mib-2 dot1dBridge(17) 11 }).

The switch implements the following traps from the mstpTraps(0) subtree of the MIB:

- The mstpNewRootBridge trap ({ mstpTraps 1 }) is generated by a bridge when it is elected as the new root of the Spanning Tree in the CIST or in any MSTI, and contains the following object:
 - mstpXstId, the MSTI or CIST instance
- The mstpNewRootPort trap ({ mstpTraps 2 }) is generated by a bridge when it changes the root port of the Spanning Tree in the CIST or in any MSTI, and contains the following objects:
 - mstpXstId, the MSTI or CIST instance
 - mstpXstPortIndex, the index of the port in the mstpPortTable table
- The mstpTopologyChange trap ({ mstpTraps 3 }) is generated by a bridge when any of its configured ports in any instance (CIST or MSTI) transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state. The trap contains the following objects:
 - mstpXstId, the MSTI or CIST instance
 - mstpXstPortIndex, the index of the port in the mstpPortTable table
 - mstpXstPortState, the new state of the port